AD-A035 753 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/6 5/1
JOINT TEST FORCE SIMULATION.(U)
DEC 76 J P EGGERS

UNCLASSIFIED GOR/SM/76D-7 NL

1 OF 2
AD A035753



AD A035753



UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
RECEIVED
FEB 18 1977
A

(11) Dec 76

(12) 183 p.

(6) JOINT TEST FORCE SIMULATION.

(9) Masters' THESIS,

(14) GOR/SM/76D-7

(10) James P. Eggers
Captain USAF

Approved for public release; distribution unlimited.

DDC
RECEIVED
FEB 18 1977
A

012 225

mt

JOINT TEST FORCE
SIMULATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

James P. Eggers, B.S.

Captain USAF

Graduate Operations Research

December 1976

Approved for public release; distribution unlimited

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

Preface

This research project is intended to provide the members of the 3246 Test Wing at Eglin AFB, Florida, with a computer simulation model which will be helpful in estimating personnel requirements and work completion dates for a joint test force.

Appreciation and gratitude are extended to Mr. Kenneth Mathiasmeier, of the 3246 Test Wing, for his interest and cooperation in determining the model requirements. His knowledge of the organization of a test was valuable in obtaining the necessary background to develop the generalized event-oriented simulation model of work activities.

Appreciation is extended to Mr. Charles Etheridge and Ms. Pam Martin of CACI, Inc., for their expertise and understanding while learning to use SIMSCRIPT II.5.

Thanks is also extended to Mr. William Drake of AFLC/XRS for his technical advice on programming a simulation model of this magnitude. His suggestions were followed without exception and proved invaluable in time savings.

Special thanks are due to Major Charles McNichols and Colonel Ronald Luhks of the AFIT Department of Systems Management. As thesis advisor and reader they provided guidance, objectivity, and expertise.

Most sincere gratitude is extended to my wife, Martha, for her encouragement throughout this endeavor and her patience during the numerous long nights required to develop this model.

JAMES P. EGGERS

Contents

	Page
Preface	ii
List of Figures	iv
Abstract	v
I. Introduction	1
Background	1
Current Knowledge	4
Previous Work	5
Statement of Problem	6
Scope	6
II. Simulation Model	8
General Concept	8
Capabilities and Limitations	10
Language Selection	12
Model Input	13
Model Description	17
Validation	36
III. Illustrative Example	38
Description of the Test	38
Data Input	38
Discussion of Output	47
IV. Conclusions and Extension of Research	49
Bibliography	51
Appendix A. Definition of Terms	53
Appendix B. Flow Chart	60
Appendix C. Program Listing	66
Appendix D. Data Input Instructions	157
Appendix E. Sample Data	163
Appendix F. Sample Output	167
Vita	174

List of Figures

Figure		Page
1	Test Cycle	2
2	Simplified Flow-Chart	9

Abstract

This study presents the background, development, and use of a generalized event-oriented simulation model of work activities. The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive? The program is written in SIMSCRIPT II.5 and is adaptable to a particular joint test force by using free-format data inputs.

Any test force which is responsible for the performance of activities may be studied using this model. Any number of activities, missions, test phases, workers, and levels of workers may be defined through the input data. Detailed input instructions are provided to tailor the model to the analyst's requirements. Numerous options are available to include: worker skill factors, mission reduction or cancellation probabilities, activity performance capability with reduced personnel, reports, parametric studies, and replications of the entire data package.

The function of each routine and event is discussed and a complete program listing is appended. A sample study is provided to illustrate many of the significant features.

I. Introduction

Background

The 3246th Test Wing, Eglin AFB, Florida, is the primary Armament Development and Test Center (ADTC) organization which provides the overall management for developmental testing. It determines ADTC test workload, identifies resources needed to accomplish testing, conducts and reports those tests for which it has been designated the responsible test organization, and supports those tests for which it has been designated the participating organization. It receives test requests from other AFSC organizations, USAF commands, DoD departments, U.S. government agencies, and allied government's national defense forces.

When a test request is received from one of these organizations, the first phase of five phases in a test cycle (see Figure 1) is initiated. Upon initial contact by a test requester, a programming engineer is designated to conduct the initial test planning. After informal contact between the test requester and the programming engineer, the test requester sends a formal request to the Test Wing; and the programming engineer prepares an initial contact summary. The Engineering Test Directorate then assigns a test analyst/test design engineer and a test engineer to design a proposed test.

A test planning conference is scheduled as a first step in drafting the work directive. The conferees determine the scope of the test, review the test objectives, and may discuss the Statement of Capability which must be prepared for accepted tests. The expected results of the conference include an outline of the test, an initial testing schedule,

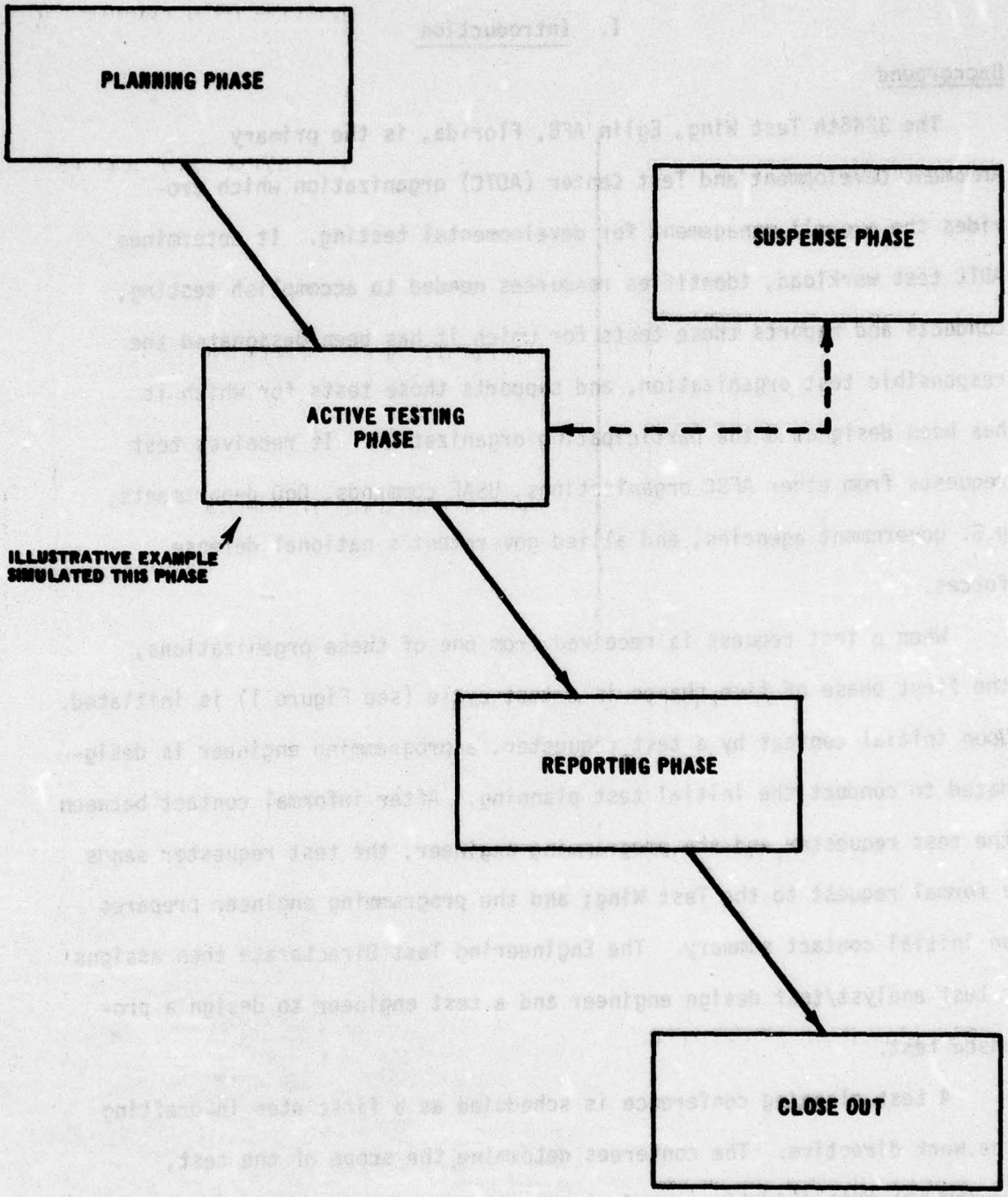


Fig. 1. Test Cycle

required facilities, and a designation of key personnel.

The test design team prepares the method of test annex which explains how the test is to be conducted and controlled, what data are to be collected, and how the data will be analyzed. Specifically, they must refine the test objectives, determine assessment criteria, identify required resources, determine environmental consequences, describe test control procedures, and estimate the data reduction and analysis effort.

After receiving the method of test, technical annexes, and support annexes, the programming engineer completes the work directive. When signed by the ADTC Commander, this directive orders the execution of a test by ADTC assigned personnel. Included in the directive are the objectives, schedule, initial priority, key personnel, and participating agencies. At this time, the planning requirements have been completed.

After completion of the planning phase several activities must be completed prior to the first mission. The test engineer must insure that the test items are available; checklists have been approved; working space for the test item contractor has been provided; and that test-bed aircraft, support aircraft, instrumentation, and radars are available for scheduling during the period of planned testing. He is then ready to submit a mission request.

When a test enters the active phase, the test engineer must be prepared to revise his testing schedule to use alternate equipment, personnel, or facilities rather than accept a delay. Delays can be expected for failures of the test item or test resources, and for scheduling conflicts because of higher priority missions. Available alternatives to scheduling delays include off-hour scheduling, accepting alternate

missions, and attempting piggyback missions with tests of higher priority. If it is determined that delays will extend the completion date, an amendment to the work directive must be initiated and a revised schedule for use of test support resources must be determined.

Tests may enter a suspense phase if a delay of at least 30 days is anticipated. A valid reason for suspending active testing might be a requirement for intermediate data reduction to help determine follow-on procedures. Secondly, the quantity of data collected may be so excessive that it cannot be properly analyzed. It may be necessary to implement improved data reduction techniques.

The reporting cycle begins on the day following the last mission, and normally 60 days are allowed to complete this cycle. Since technical reports are the chief product of the Test Wing, guidelines to include a schedule have been developed. Final data reduction and analysis is completed by the 14th day, and the draft must be completed by the 26th day. Coordination, editing, and review are completed by the 55th day; and then the Wing Commander has until the 60th day for review, approval, and signature.

The final phase of the test cycle is closeout and normally lasts from 45 to 60 days. During this period all cost accounting adjustments are finalized and letters of termination or completion are sent to all test support and other interested agencies. Thus, five phases of a test have been highlighted (Ref 3:1-44).

Current Knowledge

A recurring management problem is the allocation of scarce resources among alternative ends (Ref 15:4). Test forces which study, test, and

report the results of various Air Force projects are competing for the scarce resources of men and equipment. Inflation and current budgetary policies have effectively reduced the amount of resources available for allocation within the Air Force. Therefore, more sophisticated management tools are necessary to consider the range of available alternatives and to determine which alternative most effectively produces the desired results.

In the early 1960's simulation models, which use digital computers to provide insight into complex processes, were designed and developed. The purpose of a computer simulation is to gather information on the characteristics of a system in order to provide a method of predicting the dynamic characteristics of an organization and consequently to provide a basis for making decisions. According to Mr. Richard L. VanHorn, "Simulation offers the most flexible and realistic representation for complex problems of any quantitative technique" (Ref 18:256). Also according to George S. Fishman and Phillip J. Kiviat, "Simulation made it possible to carry out fully integrated system analyses which were generally far too complex to be carried out analytically" (Ref 6:2).

Previous Work

A Defense Documentation Center search submitted for simulation test facilities, simulation facilities, and shop scheduling systems produced no indication that a study like this one has been accomplished. A study for a similar organization, the 4950th Test Wing at WPAFB, Ohio, entitled A Multi-Project Scheduling Program, was completed in February, 1974. The problem which was modeled in the current study differs significantly from the multi-project scheduling program. The current study considered

the effects on completion dates of changing test force composition for a single project. The previous study provided the expected schedule for cycling numerous tests through an existing structure. Discussions with members of the 3246th Test Wing at Eglin AFB, Florida, also indicate that no similar studies have been performed for their organization.

Statement of Problem

An increasing number of complex tests in the 3246th Test Wing, Eglin AFB, Florida, are being performed by dedicating personnel to a joint test force (JTF). The Test Wing requested the development of an event-oriented simulation of work activities performed by the JTF and support personnel. By adapting this simulation to a current JTF, the director can estimate work completion dates, personnel requirements, and the relationship between the number of assigned personnel and the work completion dates.

The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive?

Scope

The purpose of this research was to develop a generalized event-oriented simulation model of work activities. This simulation is adaptable to a particular joint test force. Using information presented in the work directive for a test, it provides insight in estimating personnel requirements and work completion dates. Since the inputs to the simulation will be extracted from the work directive, the phases following the planning phase can be simulated.

This simulation will enable joint test force directors to estimate

the overall personnel requirements of the test force relative to a specified completion date. Secondly, they will be able to determine the relative effects of assigning various numbers and levels of workers to a test. Through the use of simulation, they will be able to identify potential problems, and experiment with various personnel configurations and work rules. Thus, by using the simulation model suitable alternatives may be found which will avoid or alleviate an indicated problem.

Since modeling resources for the entire Eglin AFB complex is beyond the scope of this study, probabilities to determine the availability of resources and activities which are not under the control of the joint test force director were subjectively assessed by the point of contact within ADTC.

II. Simulation Model

General Concept

To understand a simulation model one must note the relationship between the concepts being simulated and the simulation model. In order to determine the relationship between the number of workers assigned to a JTF and the resulting completion date, the following two items are of primary importance: What (activity) must be performed? Who (worker) will perform the activity? A simplified flow-chart showing an application of the basic concept of the simulation model is diagrammed in figure 2. A complete flow-chart for the simulation model is listed in Appendix B. For illustrative purposes, the active testing phase of a test cycle was simulated.

Activity/Mission/Test Phase. In order to satisfy the requirements of the test objectives developed during the planning phase, certain actions (activities) must be accomplished. The work directive specifies a sequence of actions which must be completed to satisfy each objective. These sequences of actions are called missions. Frequently, a mission must be completed several times or with slight variations in order to satisfy the objectives. In the simulation model, this completion of similar missions is called a test phase. A test phase specifies the number of similar missions which must be completed.

An activity is the basic building block in the actual test as well as in the simulation model. A common characteristic of all activities is that a quantity of time must elapse prior to completion. An analyst will define the activities to match the level of detailed desired

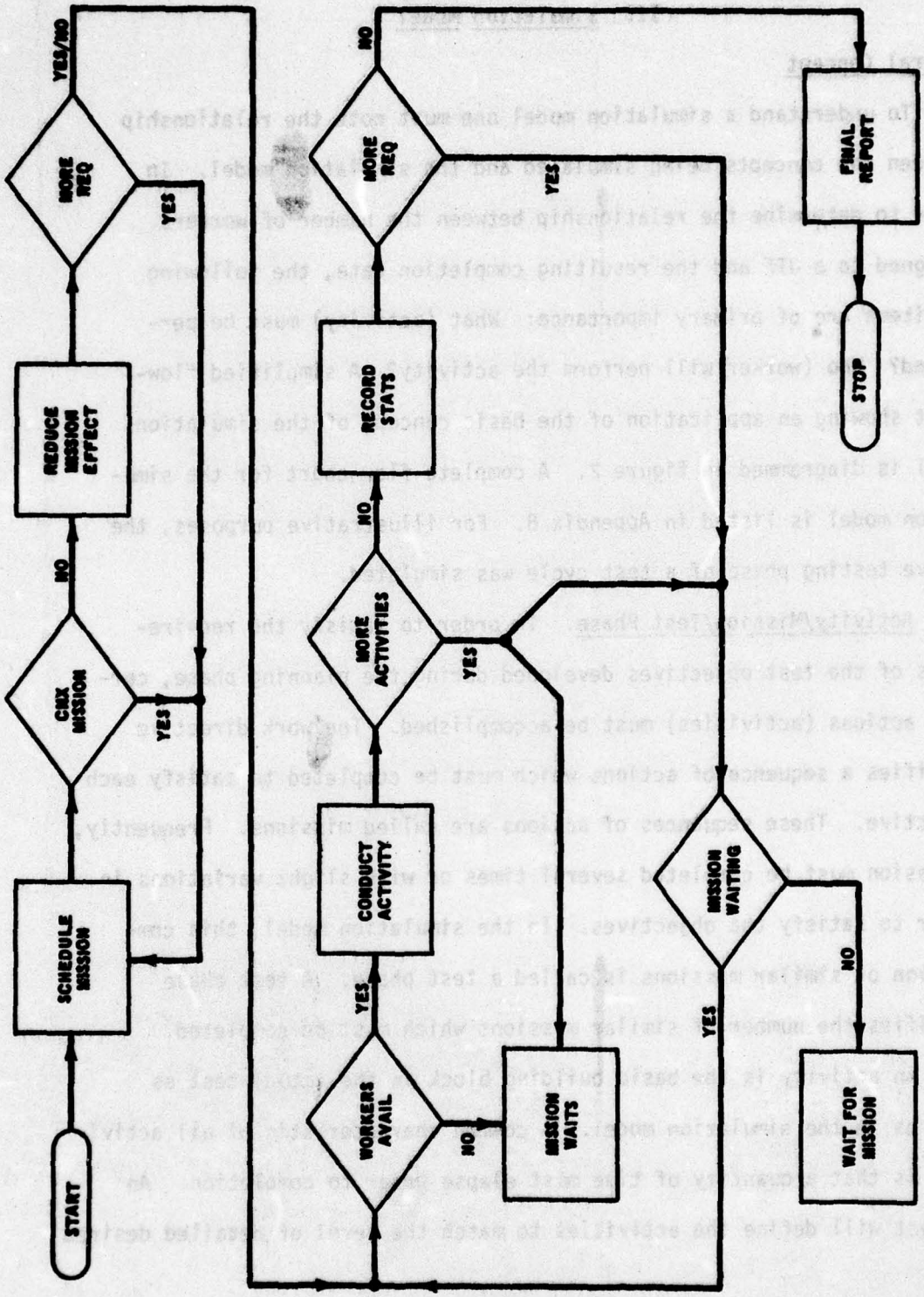


Fig. 2. Simplified Flow-Chart

for modeling a test. In the illustrative example, 14 activities were defined. At this selected level of detail, "conduct an air mission" was designed as an activity. This portion of a mission could have been modeled at a greater level of detail by defining activities such as drive to the hangar, become airborne, fly to the range, make twelve passes over a target, etc.

Using this simulation model, a test can be simulated at whatever level of detail that the inputs can be accurately determined. The same test can be simulated at different aggregate levels depending on the desired accuracy of the output and the level of effort that one is willing to expend in developing a data package for the simulation model.

Worker. The second major modeling concept is: Who (worker) will perform the activity? Different levels of workers are defined to accommodate the idea that certain activities are accomplished by only a certain type of person. In the simulation model workers may be stereotyped into different levels of workers. The analyst may relate these levels to such items as job titles, AFSC's, etc. Each worker may be assigned a skill factor for each activity. This factor adjusts the activity time to reflect the realization that all people do not perform an activity in the same length of time. These two concepts form the basis of the model. Definitions are provided in Appendix A to clarify the meaning of the various simulation terms that are used in discussing the model.

Capabilities and Limitations

Capabilities. Any test force which is responsible for the perform-

ance of activities may be studies using the model developed in this research effort. Any number of activities, missions, test phases, and workers may be defined through the input data. Activities may be grouped in any order to represent a mission in a particular test phase. Additional flexibility is provided by allowing worker requirements, activity sequencing, and mission priorities to be defined through the input data. Situations in which there are no similarities between missions can be represented at the expense of the additional data inputs to separately describe each mission.

Three worker level preferences may be specified for each activity. The selection of members from these levels affects the activity time, by averaging the skill factors of the assigned workers. After team assignments have been made, the remaining qualified workers are assigned to approved activities on a partial-team basis. The selected activity time is increased in proportion to the number of workers required. Activity times may be specified for each test phase and, consequently, reduce the number of separately defined activities. Any number of cancellation factors may be entered for each mission, and each activity has the capability of reducing the mission effectiveness. Numerous options are listed under Options in the Model Input section.

Limitations. A simulation model is an effective method for studying complex systems. However, a model should not be used without a thorough understanding of the meaning of each required input. The model does not refuse to give answers simply because the inputs are inaccurate or unreasonable. Since there is a certain amount of

abstraction in collecting or constructing any data package, there exists a possibility of biasing the results. In a stochastic model, the sensitivity of these inputs should be tested to insure that the estimates do not seriously bias the results.

Language Selection

Selection of a simulation language is determined primarily by the nature of the problem, and secondly by the accessibility of the various languages. Different languages are designed to model systems involving a flow of discrete items. The problem considered is one of discrete items. Discrete-model languages perform the following functions: advance the simulated time, search for the next scheduled event, place a new event on the event file, and test for status matches to trigger conditional events (Ref 5:142).

The two most widely used event-oriented simulation languages are General Purpose Simulation System (GPSS) and SIMSCRIPT. GPSS automatically performs the following functions: collects data describing a model's simulated behavior, prints summaries of the data at the end of the simulation, and provides the format statements indicating how the data summaries are to be displaved. GPSS has incorporated much of the underlying logic of discrete event simulation, and thereby permits a model builder to use the language without understanding its internal logic. Use of a high level language in this manner invariably results in model invalidities which are frequently subtle in nature, and therefore are difficult to detect (Ref 16:vii).

SIMSCRIPT is a general language which can do everything that can

be done in FORTRAN. The model builder uses named entities to represent objects in the system and has complete freedom to define the properties of these entities. SIMSCRIPT also has an extensive report-generating feature which displays the selected output in a prescribed format. Although SIMSCRIPT performs the basic functions of event-oriented languages, it requires an analyst to translate the problem formulation into appropriate model representation and provide some of the programming details. (Ref 17:214)

SIMSCRIPT II.5 was selected for this study for several reasons. According to George S. Fishman and Phillip J. Kiviat, "Very minor simplifying assumptions can generate output processes whose behavior differs considerably from what is desired." (Ref 6:14) Since SIMSCRIPT II.5 is a full programming language, the number of assumptions necessary in modeling the test force were minimal. Also since every statement in every routine is written by the programmer, there is no guesswork involved as to the function of a particular segment of the program. Finally, the extensive report generating feature of SIMSCRIPT II.5 permitted formatted reports tailored to meet the requirements of weekly, monthly, and final reports.

Model Input

Instructions. Detailed data input instructions are listed in Appendix D. However, prior to answering these questions, the basic structure of a test must be determined.

Activities. First the required activities must be identified from the required resources for each mission and the test objectives.

Criteria used to determine the number of different kinds of activities that need to be identified is generally based on the required equipment and the degree of difficulty. Specific questions that must be considered include:

1. Will the activity be performed during normal duty hours?
2. What type of distribution will describe the time required to perform the activity?
3. What levels of workers can perform this activity?
4. Can the activity be expected to reduce the mission effectiveness?
5. By what percentage is the activity expected to reduce the mission effectiveness?

After defining these basic characteristics of the activities, the activities are further defined by entering the time function parameters and the number of workers required in each test phase. Thus, the data input requirements have been reduced since the basic activity data are not repeated for each test phase.

Missions. Secondly, based on similarities in activity requirements and timing requirements, missions must be constructed for each test phase. Normally, missions in a test phase consist of a standard sequence of activities; however, minor differences can be accommodated by using external event notices to create a mission. External missions select a subset of the standard activity sequence, and also permit an activity to be added to the sequence by entering the required information. By using this feature, the number of test phases required

may be reduced.

Workers. Thirdly, the categories or level of workers must be determined based on the requirements for each activity. The actual number of workers in each level and their skill factors in performing each activity can be varied for each simulation run. The workers are a primary variable in determining the completion date. To minimize data input, the default of the same skill factor for all workers is recommended for the initial runs.

Program Variables. After the basic structure is determined, entries regarding quantities, times, probabilities, priorities, programming options, and programming control variables are entered. Names of workers and activities should be no more than six characters.

Activity Test Phase Codes (ATPC). Activity test phase codes determine the order in which activities will be performed. They are composed of two segments and an optional prefix to provide priorities between missions of different test phases. The first segment corresponds to the test phase numbers which are numbered chronologically from 1 to the total number of test phases. The second segment indicates the order in which activities will be performed within the test phase. All codes must have the same number of digits.

Mission Priorities. Normally it is anticipated that the priorities between missions of different test phases are implied priorities corresponding to the chronological order in which they are scheduled. In this situation no prefix is added to the test phase

number in the activity test phase code. To explicitly assign priorities between missions in different test phases, all activity test phase codes must be prefixed with a priority number. (Low number indicates the highest priority.)

For example, the first activity to be performed in each of the first four test phases would be coded 11, 21, 31, and 41. To assign a higher priority to test phase 3 missions, the ATPC's would be prefixed as follows: 211 221, 131, 241.

Empirical Distributions. Empirical distributions may be entered for activity times which cannot be accurately described by one of the ten provided distributions. It is recommended that the use of this feature be minimized. According to George S. Fishman and Philip J. Kiviat, "Sampling from a theoretical rather than empirical distribution is generally considered preferable, since it exposes a simulated system to the universe of possible stimuli rather than merely to those that have occurred in the past" (Ref 6:20).

Options. Numerous yes/no options are provided in the input data to tailor the model to the analyst's requirements. Depending on the total length of the test and the details required, weekly and monthly reports may be omitted. The features which reduce mission effectiveness and provide mission cancellations may also be deleted. The user can specify the number of workdays per week, and also may elect to work priority missions on non-duty days. On initial runs the same-skill-factor option, which assigns a value of 1 to every worker for each activity, may be used. External events may be used

to schedule a specific mission on a certain day or to terminate the simulation on a specific date. Finally, replications of the entire data package may be run either without resetting the random number generators or by entering new seed values for the random number generators.

Model Description

This section describes the function of each routine and event in the simulation model. The descriptions are presented in the same order as they are listed in the program in Appendix C. This order also corresponds to the general sequence in which the events and routines are used when conducting a mission. The reader should scan this section during the first reading, and then refer to a particular routine or event description to better understand some of the ideas presented later. If more detail is required to clarify a point, the reader is directed to the program listing in Appendix C.

PREAMBLE. The PREAMBLE declares the static system structure of the model, using an entity-attribute-set framework (Ref 12:193-207); declares the events that compose the dynamic structure; defines special properties of the two structures, such as the mode of attributes, the ranking of sets, and priority order of events; and specifies data-collection and analysis tasks. This set of global declarations describes to the compiler the system being simulated and the data-collection and analysis statements.

The static system structure includes the permanent entities, empirical functions, temporary entities, and the system-owned sets.

The permanent entities are TEST.PHASE, KIND.OF.ACTIVITY, LEVEL.OF WORKER, and WORKER. A TEST.PHASE represents a requirement that a certain number of similar missions be completed. Its attributes record information regarding numbers of missions. A KIND.OF.ACTIVITY is created for every different type of activity, and has attributes which identify the parameters of each activity. A LEVEL.OF.WORKER is created for each category of workers and is used to gather utilization statistics. A WORKER is created to represent each person on the joint test force. Its attributes identify the person, his capabilities, and his current status. Compound permanent entities are TEST.PHASE/KIND.OF.ACTIVITY and WORKER/KIND.OF.ACTIVITY: activity duration times in each phase, and the worker skill factors for each activity are stored in the compound entities.

Three types of empirical functions can be entered for activity times which cannot be characterized by one of the ten provided standard functions. When using an empirical function, a sample value is selected by generating a random number, matching it against the probability entries, and thus selecting the activity time. INTERGER.STEP.FUNCTION and REEL.STEP.FUNCTION select in a step-like manner a real or integer value. CONTINUOUS.FUNCTION samples with linear interpolation between real sample values. Sampling is performed by successively comparing a random probability value to the cumulative probability value until a value is found that is greater than or equal to it. The associated value or the interpolated value between the associated value and the value preceding it is returned as the sample

value for step and continuous functions respectively.

The temporary entities are MISSION, ACTIVITY, AND XFACTOR. When a MISSION is created as a result of an internal event notice, the sequence of activities is copies from the set called STRUCTURE of the designated TEST.PHASE, STRUCTURE(TEST.PHASE), into the set called ROUTING, which is owned by the newly created MISSION, ROUTING(MISSION). If the MISSION is created as a result of an external event notice, the selected subset of activities is copies into ROUTING(MISSION). Additionally, activity codes which are not a part of the standard sequence may be entered with the required attributes. When such an entry is made, a warning, along with the necessary information to verify the input, is printed.

Each XFACTOR represents a reason for which a scheduled mission could be cancelled. These factors are used to represent items which could be expected to affect both standard and special missions. Examples are cancellation because of a higher priority mission and unsuitable weather.

The system owns two sets in which missions are filed until certain conditions exist. The members of the set called QUEUE are all the missions that are waiting for workers to perform the next activity. Missions are filed in the QUEUE on a first in, first out basis ordered by low priority code, then by low activity code, and then by low notification time of the impending activity. The set called WAIT.FOR.MISSION.TIME has as its members all missions for which all pre-mission activities have been completed prior to the scheduled active mission time. Missions are filed in WAIT.FOR.

MISSION.TIME on a first in, first out basis ordered by low notification time.

The dynamic system structure describes the time-dependent behavior of the system being simulated through the use of event notices. An event is an instant in time at which an activity starts or stops; it is executed in zero simulated time; and it changes the state of the system.

The event notices provide the proper sequencing of activities so that the order of performing activities in the model corresponds to the system which is being modeled. The periodic events are discussed in the sections entitled MAIN and REPORTS. Other events are dependent on the state of the simulation for scheduling and are discussed in the order that they are listed in the program. The capability of scheduling a special mission or of ending the simulation on a specific date from outside the simulation model is provided through the use of external event notices.

The Data Collection and Analysis declarations specify which statistics are to be gathered during the simulation. These statistical measurements are the principal output of the simulation experiments. Through the use of several ACCUMULATE and TALLY statements in the preamble, the operating program is free of data collection and data reduction statements. The ACCUMULATE statements gather time dependent statistics on mission waiting time and worker utilization. The TALLY statements compute various statistics associated with mission duration times.

The miscellaneous declarations define several abbreviations and global variables used in the program. Numerous functions are defined

in this section which are used for simulation control.

MAIN. The MAIN routine provides overall simulation experiment control. It calls two programmer-written routines to initialize the static system state, and it schedules events for the system-provided timing routine. The START.SIMULATION statement removes the first scheduled event from the system file of scheduled events and starts the simulation by transferring control to it. Tracing of event notices is provided during the tracing period by calling MYE.TRACE.

Eventually, either by running out of data or by programmer action, control passes from the timing routine to the statement following the START.SIMULATION statement. If there is no more data, the simulation is terminated. If there is more data, the system is initialized for another run. If more than one run of the entire data package is requested, routine REPLICATION is called to reset the simulation model.

In MAIN several SCHEDULE statements specify those events which will occur periodically throughout the simulation. When a SCHEDULE statement is executed, an event notice of the specified type is put in a file ordered by scheduled time; when the simulation clock advanced to this time, the event is executed. The events BEGIN.WORKING,TAKE.BREAK, and EO.DUTY.HOURS call routines to assign or remove workers from the activities of the current missions. Depending on the priority of the mission and current simulation time, worker status is changed to reflect the current conditions. The first CONFIRMED.MISSION in test phase 1 is scheduled for 0800 on the first day of the simulation. The weekly report and monthly reports are optional and will be discussed along with the final report in the reports section. The routines INITIALIZE,

REPLICATION, XWREPORT. and XMREPORT are released when the data indicates that they are not required, and thereby provide additional memory to run the simulation.

INITIALIZE. The INITIALIZE routine, which is called in MAIN, reads all data that will remain constant for the duration of the simulation run. These data include information relevant to the different test phases, missions, activities, time functions, and simulation control variables. The date reference is established by calling the system routine ORIGIN.R. Storage is allocated for the entities by using the CREATE statement, and then the various attributes are read. A "master copy" of a mission in each test phase is created by sequentially filing activities in the set owned by each test phase. The correct flow of input data is checked at several points by inserting the character string "OK". If the data input is incorrect, the run terminates; and an error message is printed which indicates where the error was detected. Data which are correctly entered is then printed for verification and reference when analyzing the simulation output. Upon return to MAIN for the final replication of the entire data package, the space occupied by the routine is released and may be used for dynamic storage such as creating and scheduling temporary entities and event notices.

RUN.INITIALIZATION. The RUN.INITIALIZATION routine is called in MAIN for each run and facilitates parametric studies. Input data for each current run are used with the background data read in INITIALIZE to provide the data package for the current simulation run. The data include information relevant to the workers, levels of workers, skill factors, and the number of days required between missions. The correct flow in input

data is again checked and printed as in INITIALIZE. the space occupied by this routine is not released and is used for each run until the data are exhausted.

BEGIN WORKING. The event BEGIN.WORKING is scheduled at the beginning of each workday and at the end of each break. It changes each worker's status to idle if he is on break or off-duty, and too busy if he is working overtime. It then calls the routine CHECK.QUEUE to see if there are any missions to be performed.

CHECK.QUEUE. The routine CHECK.QUEUE is called from the BEGIN.WORKING, CONFIRMED.MISSION, EO.ACTIVITY, ACCOMPLISH.PRI.MISSION, EO.DUTYHOURS, and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. Every mission which has completed its premission activities prior to mission time is removed from QUEUE and filed in WAIT.FOR.MISSION.TIME. If the current simulation time is during nonduty hours, either after duty hours or on the weekend, the routine OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is called. If the check is made during duty hours, ATTEND.TO.MISSION is called for each mission for which there is a sufficient number of qualified workers. During normal duty hours, BUILD.PARTIAL.TEAMS is then called to try to assign the remaining workers.

CONFIRMED.MISSION. The event CONFIRMED.MISSION is notification of a scheduled mission. It is initially scheduled for 0800 of the first day of the simulation and is subsequently scheduled as directed in the input data. When notification is received, DISBAND.PARTIAL.TEAMS is called to free team members for new assignments. Notification of an internally scheduled mission is accepted during nonduty hours, but not on nonduty days (the number of duty days is a user input). Rejected missions are

rescheduled for the same time the following day. Externally scheduled missions are accepted at any time.

If the event is scheduled internally, a mission is created with the standard activities, effectiveness reductions, and cancellation factors for a mission in the indicated test phase. If the notification was received through external event notice, the indicated subset of activities and any additional activities are read as data input. A mission is then created with these required activities and the associated effectiveness reduction and cancellation factors. For each new mission VALIDATE.THE.NUMBER.OF.WORKERS.REQUIRED, DETERMINE.PREMISSION.CANCELLATION, and CALCULATE.MISSION.EFFECTIVENESS are called. If the mission is cancelled, BUILD.PARTIAL.TEAMS is called. If the priority of the first activity is 1, an ACCOMPLISH.PRI.MISSION is scheduled. Otherwise, depending on the simulation time, ATTEND.TO.MISSION or OVERTIME.ACTIVITIES.FOR.OVERTIME.MISSIONS is called. BUILD.PARTIAL.TEAMS is called to try to assign the remaining workers. LOOK.AHEAD.COMPLETE.MISSIONS is called to determine the test phase of the next mission, followed by MISSION.SCHEDULE.

MSN.SCHEDULE. The routine MSN.SCHEDULE is called by CONFIRMED.MISSION to schedule the arrival of the next mission. If the number of days between missions is less than 1 and the computed time is within normal duty hours, the mission is scheduled for that time. If the computed time is after duty hours, the mission is scheduled for the beginning of duty hours on the next duty day. If the number of days between missions is 1 or more, the mission is scheduled in that number of days at a time selected randomly from the user-provided function for the time

of day.

VALIDATE.NUMBER.OF.WORKERS.REQUESTED. The routine VALIDATE.NUMBER.OF.WORKERS.REQUESTED is called by CONFIRMED.MISSION. It checks each activity in a mission to insure that the number of workers requested in each level have been provided in the simulation. If the request is greater than the number of qualified workers in the simulation, an error message is printed with information describing the current state of the simulation. The simulation is then terminated. This check is necessary to preclude generating a mission which could never be completed because its requirements could never be met. This error in input data is normally encountered when conducting parametric studies.

DETERMINE.PREMISSION.CANCELLATION. DETERMINE.PREMISSION.CANCELLATION selects a random probability for each cancellation factor of a mission. If the selected random probability is less than the corresponding factor, the mission cancellation code indicator is set equal to 1.

CALCULATE.MISSION.EFFECTIVENESS. A similar function is performed by the routine CALCULATE.MISSION.EFFECTIVENESS. For each activity of a mission, a random probability is selected and compared to the percent of missions affected. If the selected value is less than this percentage, current mission effectiveness is then reduced using a uniform distribution between the upper and lower limits indicated for each activity. If the total reduction exceeds 100 percent, an informational message is printed and the cancellation code indicator is set equal to 1. This action was programmed to prevent a possible situation in which the workers would always be busy but never complete a productive mission. It is assumed that, if this observation were made at an installation,

corrective action would be taken. Upon return to CONFIRMED.MISSION, missions with cancel code equal to 1 are cancelled and a new mission is scheduled.

ATTEND.TO.MISSION. The routine ATTEND.TO.MISSION is called only during normal duty hours by the CONFIRMED.MISSION, ACCOMPLISH.PR1. MISSION, or CHECK.QUEUE. Every priority mission which has completed its premission activities prior to mission time is removed from the QUEUE and is filed in WAIT.FOR.MISSION.TIME until its mission time (time of notification plus a user provided mission preparation time). For all other missions in which the number of available workers qualified to perform its next activity exceeds the number of workers required for that activity, the routine ALLOCATION is called.

SUM.WORKERS. In determining the number of workers currently available to perform an activity, the routine SUM.WORKERS considers all combinations of worker level preferences. Depending on the input arguments, a combination of the four sections is used.

The first section determines the number of idle workers. If the activity has a priority 1, the fourth section adds the additional number of qualified workers who are not working on another priority 1 activity.

The second section determines the number of qualified workers who are not working. This section is primarily used for priority 1 missions which arrive during break times. Since workers who are busy during a break are by definition working on a priority 1 mission, the qualifications of the remaining workers are tested.

The third section also determines the number of qualified workers who are not working. This section is used for overtime missions which

may be either priority 1 or priority 2 missions. By using the second section and a local variable indicating overtime, the number of qualified workers not working overtime is determined. If the activity has a priority of 1, the fourth section adds the number of qualified workers which are not working on another priority 1 activity. Additions made in the fourth section will be made available as the priority 1 activity preempts lower priority activities to meets its requirements.

In all cases, the routine returns the number of qualified workers which is then compared to the number of workers required to perform an activity. If the required number of workers is available, the routine ALLOCATION is called.

ALLOCATION. The routine ALLOCATION is called by ATTEND.TO.MISSION and assigns qualified workers to a team for activities which are performed during normal duty hours. If the number of required workers is zero, indicating that the manpower requirements are external to the test force, an activity time is selected. Otherwise, team members are assigned in the order of the activity's worker level preference. Selection within a level is based on the least busy time for each worker. If the activity has a priority 1 and there are not enough idle workers to fill the team, lower priority missions are preempted until the required workers are found. After the team assignments are made, an activity time is selected and an EO.ACTIVITY is scheduled. Upon return to the original calling routine, CHECK.QUEUE is called to reassign workers who may have been preempted.

ACCOMPLISH.PR1.MISSION. The event ACCOMPLISH.PR1.MISSION is scheduled for priority 1 activities at a user-supplied length of time after

notification of a scheduled mission. This time represents the required preparation time before conducting the first priority 1 activity in a mission. DISBAND.PARTIAL.TEAMS is called to free team members for new assignments. Dependent upon the current simulation time, ATTEND.TO.MISSION or OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is called. CHECK.QUEUE is then called to reassign workers if a preempt occurred.

PREEMPT. The routine PREEMPT is called by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. PREEMPT selects one non-priority 1 mission to preempt each time it is called and then calls INTERRUPT.AND.FILE.

INTERRUPT.AND.FILE. The routine INTERRUPT.AND.FILE is called by PREEMPT, TAKE.BREAK, and EO.DUTY.HOURS. It removes the workers from the team, calculates the remaining activity time and files the interrupted mission in the QUEUE.

VALUE. Routine VALUE is called by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. If the activity has previously been interrupted, it multiplies the continuation time by the average skill factor of the new team members and the partial team factor. It then returns to the calling routine. Otherwise, it selects an activity time using one of the ten provided distributions or a user provided empirical distribution. The arguments for the function are indicated for the activity and test phase. A sample value is selected from the function and this value is multiplied by the average skill factor of the workers assigned to perform the activity and the partial team factor. The weighted time value is returned as the activity time.

LOOK.AHEAD.COMPLETE.MISSIONS. The routine LOOK.AHEAD.COMPLETE.MIS-

SIONS is called by CONFIRMED.MISSION. It sums the final mission effectiveness for all test phase missions which are currently in the simulation. This value is added to the number of completed missions in a test phase and the total is compared to the number of required missions. If the total exceeds the number of required missions, a mission in the next test phase will be scheduled. This routine was developed on the assumption that prior to completing the final activity in a mission, an approximate mission effectiveness can be determined. In a simulation run in which the number of missions in the QUEUE builds or the mission duration is lengthy, an unrealistically excessive number of missions would be scheduled in a test phase. The look ahead routine is provided to allow the simulation model to perform like the observed system.

TAKE.BREAK. The event TAKE.BREAK is scheduled periodically at the user-provided times for coffee and lunch breaks. It puts all idle workers on break status, identifies every non-priority 1 mission in progress, and calls INTERRUPT.AND.FILE for each of these missions.

EO.ACTIVITY. The event EO.ACTIVITY is scheduled by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS for each activity at the simulation time determined in routine VALUE. It calls DISBAND.PARTIAL.TEAMS to free team members for new assignments. It then removes the completed activity from the set of activities required for the mission, ROUTING (MISSION); changes the mission priority and mission code to correspond to the next activity (determines ranking order in QUEUE); and removes the workers from the team (activity). Worker status is then determined based on current simulation time. The mission is then filed in the

QUEUE insuring that low coded activities of all missions are performed first. If this event is called during a weekday or if the user has allowed priority missions to be worked on the weekends, CHECK.QUEUE is called.

EO.DUTY.HOURS. The event DO.DUTY.HOURS is scheduled daily at the user-provided time of day. It calls DISBAND.PARTIAL.TEAMS to stop the associated activities. The status of every idle worker is changed to off-duty, and the status of team members of priority 1 or priority 2 missions is changed to overtime. Regardless of priority, missions with less than 30 minutes remaining on the current activity will be completed on overtime. This action is based on the assumption that a worker will elect to complete a task that is nearly finished so that the following day's efforts can be devoted to new activities. OVERTIME.ACTIVITIES.FOR. PRIORITY.MISSIONS is then called to work any priority missions which are waiting in the QUEUE.

OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. The routine OVERTIME. ACTIVITIES.FOR.PRIORITY.MISSIONS is called by CONFIRMED.MISSION, CHECK. QUEUE, ACCOMPLISH.PRI.MISSION, and EO.DUTY.HOURS. It is used for missions performed during non-duty hours throughout the work week, and on weekends if the user has allowed priority missions to be worked on the weekends. Every mission which has completed its premission activities prior to mission time is removed from the QUEUE and filed in WAIT.FOR.MISSION. TIME. If the number of required workers is zero, indicating that the manpower requirements are external to the test force, an activity time is selected. Otherwise, team members are assigned in the order of the activity's worker level preference. Selection within a level is based

on the least overtime for each worker. If the activity has a priority 1 and there are not enough off duty workers to fill the team, lower priority missions are preempted until the required workers are found. After the team assignments are made, an activity time is selected and an EO.ACTIVITY is scheduled. If the activity is initiated on a weekend, an informational message is printed regarding that activity. Upon return to the calling routine, CHECK.QUEUE is called to reassign the workers if a preempt occurred.

BUILD.PARTIAL.TEAMS. The routine BUILD.PARTIAL.TEAMS is called by CHECK.QUEUE and CONFIRMED.MISSION after workers have been allocated to the activities. This routine selects those missions whose impending activity can be started with a partial-team selected from the remaining workers. ALLOCATION is called for each of these missions.

DISBAND.PARTIAL.TEAMS. The routine DISBAND.PARTIAL.TEAMS is called by CONFIRMED.MISSION, ACCOMPLISH.PRI.MISSION, TAKE.BREAK, EO.ACTIVITY, AND EO.DUTY.HOURS. These teams are disbanded during breaks and any time a new activity comes available for worker allocation. Thus, activities for which the required number of workers are available will receive worker assignments prior to making any partial allocations. Partial-teams are not permitted to work on activities during breaks or nonduty hours.

COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION. The routine COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION is called by the PREAMBLE prior to destroying the record of a mission. If the cancel code indicator is equal to 1, that value is added to the number of cancelled missions in the test phase. Otherwise the mission duration is computed and the cur-

rent effectiveness level is added to the number of completed missions in the test phase. If the number of completed missions in each test phase is greater than or equal to the number required, an EO.SIMULATION is scheduled.

REPORTS. The events WREPORT and MREPORT are scheduled weekly and monthly, respectively, if requested by the user. Otherwise, these events are not called and the releasable routines XWREPORT and XMREPORT which contain the actual reporting program, are released (events are not releasable). The FINAL.REPORT is always printed at the end of each simulation run.

The same format is followed in all three reports with only minor variations in the information presented. Since there is a variable number of days in a month, certain values are presented only in time units in the monthly report rather than as percentages. The weekly and monthly reports print the cumulative number of completed missions in each test phase. The final report prints the date that each test phase was completed and the ending date of the simulation.

There are three major sections in each report: mission statistics for each test phase, waiting time information for all missions, and worker utilization information. The mission statistics presented for each test phase include the number scheduled, number cancelled, number required, effective productivity, cumulative number completed, average mission duration, standard deviation of duration times, total of duration times, sum of the squares of duration times, and the date the test phase was completed.

Waiting time information for all missions is presented in the queue

report. The information presented includes the maximum, minimum, average, standard deviation, and histogram of the number of missions waiting for workers. Also, the total waiting time for all missions is reported. This information is not printed by test phase because there is not necessarily a correlation between the waiting time for missions in a test phase and the requirements of that test phase. Since all missions fill worker requirements from the same groups of workers, overlapping of test phases, differing priorities and numerous other factors may cause missions in a test phase to wait for an untractable variety of reasons. The significant information to be derived from this section is the time frame in which large backlogs of work can be expected and the relative size of the backlog. The cure for these anticipated problem areas is indicated in the final section of the report.

Worker utilization information is presented in the third section of the report. Information includes a histogram of worker's idle time by worker level and the report of each worker's status divided into the categories of idle, busy, break, off duty and overtime. Types of information which can be extracted from this data might include: which worker levels can tolerate or require personnel changes, base figures for estimating the value or cost of these changes, and an indication of the amount of overtime that can be expected with various configurations of the simulation. Thus, this section indicates areas in which changes could be made to alleviate those problem areas highlighted in the first two sections.

LARGEST. The routine LARGEST is called by each of the reports to determine the length of the worker idle time section. It checks all

levels of workers for the maximum number of idle workers in the reporting period and returns the largest value to the calling routine.

MY.TRACE. The routine MY.TRACE is called by every routine and prints information only during the tracing period designated in the data input. It prints the name of the called routine, current simulation time, and the associated mission number.

MYE.TRACE. The routine MYE.TRACE is called by every event and prints information only during the tracing period designated in the data input. It prints the name of the called event, current simulation time, and selected information.

MY.LIST. The routine MY.LIST is called as required throughout the simulation to print the members of sets and the attributes of permanent or temporary entities.

QENTER.INFO, QEXIT.INFO. Each time a mission enters or exits the QUEUE, the routine QENTER.INFO or QEXIT.INFO is called by the PREAMBLE. During the tracing period, these routines then call INFO.PACKAGE to print relevant information.

WENTER.INFO, WEXIT.INFO. Each time a mission enters or exits WAIT.FOR.MISSION.TIME, the routine WENTER.INFO or WEXIT.INFO is called by the PREAMBLE. During the tracing period, these routines then call INFO.PACKAGE to print relevant information.

ERROR.PACKAGE. The routine ERROR.PACKAGE is called whenever a programmer detected error in the simulation is found. These messages are generated independently of the SIMSCRIPT error messages and indicate conditions which may not be detected as errors by the compiler. Four available error messages are printed as required with the appropriate

information to analyze the detected error.

INFO.PACKAGE. The routine INFO.PACKAGE is a collection of statements to list information describing the current state of the simulation. It is called from numerous locations throughout the program.

EO.SIMULATION. The event EO.SIMULATION may be scheduled internally or externally; and subsequently, it schedules the FINAL.REPORT. Internally, it is scheduled by COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION whenever the required number of missions in each test phase has been completed. If the event is scheduled externally, information describing the current state of the simulation is printed prior to scheduling the FINAL.REPORT.

CLEAR. The routine to CLEAR is called by the FINAL.REPORT to reset the state of the simulation for successive experiments. The starting conditions of the model are identical to the previous run to include restoring the seed values of the random number generators. Using this routine, a series of sequential runs with slightly modified versions of selected input can be examined. The principal use is to measure the behavior of alternative inputs in the form of number of workers, skill factors and days between missions. By using this routine, parametric studies can be conducted without repeating an entire data package.

REPLICATION. The routine REPLICATION is called if the user requests more than one run of the entire data package. This routine clears the model structure which was created by the previous set of data. However, seed values of the random number generators are not reset to the starting conditions. Optionally, the user may enter new seed values. This simulation model is designed to study the transient

condition, represented by starting the simulation with no missions in the system and continuing until the required number of missions have been completed. In order to determine a confidence interval, the run must be replicated to obtain a distribution of results. This routine allows multiple replications with different seed values for the random number generators.

Validation

The problem of verifying computer models remains the most elusive methodological problem associated with computer simulations (Ref 13:B92). A simulation is only an abstraction of the process under study and because simulations tend to become so complex that they conceal their assumptions and processes (Ref 18:247). Thus, validation is necessary to increase to an acceptable level the confidence that an inference about a simulation process is correct for the actual process. The following steps are necessary in attempting to validate a simulation model (Ref 7:1004):

1. Individually test the subroutines.
2. Review in detail the program logic.
3. Review all system state changes during the test runs under various operating policies.
4. Compare the results of the model to historical data.

Initial checkout runs started with very simple data in the form of 2 activities, 1 mission required, 1 test phase, 1 worker, and constant activity times. The complexity of the data was then increased to verify the different features provided in the program. After validating the model in a strictly deterministic mode, uniform distri-

butions with very small ranges for activity times were added. The results were nearly the same as for the deterministic runs.

The programmer written trace routines were designed specifically to present the information necessary to test the routines and to observe system state changes. Each time an event or routine is called during the tracing period, the appropriate information is printed to show changes in worker status, mission status, or system statistics. If necessary, every mission can be traced through the entire simulation to verify that every routine operates properly under all operating conditions.

After numerous runs with test data constructed to detect any possible errors in logic, a set of data was developed for the Test which was studied to provide guidance in developing the simulation model. Although all phases of the Test have yet to be completed, computer results compare favorably with test phase completion dates, worker utilization, and the time frame of backlog buildups. Thus the model structure was further verified by introducing complex data of an actual Test and by comparison with current historical data.

III. Illustrative Example

Description of Test

A current test, Pave Tack Night Attack Systems/F-4E, is used to illustrate some of the significant features of the simulation model and to show how a joint test force director can adapt this model to a particular test. This test is an integrated Developmental Test and Evaluation/Initial Operational Test and Evaluation (DT&E/IOT&E) effort under the direction of a Joint Test Force (JTF). The major objective of this test is to provide higher headquarters with sufficient information to support a production decision on the Pave Tack Class V Modification to the F-4E aircraft. Sufficient data are required on competing systems to support a source selection decision. (Ref 2:4)

For this example, the DT&E portion was simulated since a significant portion of that test has been completed and was referenced for validation of the simulation results. The estimated active phase of the test is between 25 May 1976 and 28 February 1977. The objectives and required resources were extracted from the work directive with the help of the ADTC point of contact. Pave Tack personnel were consulted to verify and refine the time estimates. A total of 66 missions were simulated.

Data Input

Activities. Fourteen activities that were based on the required resources and the test objectives were identified after analyzing the work directive. These activities are as follows:

1. PMSNPG - Premission preparation and briefings for a ground mission

2. PMSMPA - Premission preparation and briefing for an air mission
3. CGDMSN - Conduct a ground mission
4. CAMSN - Conduct an air mission
5. PSTMNG - Post-mission evaluation for a ground mission
6. PSTMNA - Post-mission debriefing for an air mission
7. DATARD - Data reduction associated with an air mission (No JTF personnel required)
8. DATAAN - Data analysis for an air mission
9. PMSNI - Premission preparation and briefing for an integration test
10. CMSNI - Conduct an air mission for an integration test
11. PSTMNI - Post-mission evaluation for an integration test
12. MSNNEL - Mission performed at Nellis (TDY for JTF personnel)
13. ANTAPE - Analyze tapes
14. ANLOGS - Analyze logs

The definition of these activities can be kept general since activity times and number of required workers are entered for each test phase in which the activity is needed. A defined activity may be used more than once in a single mission if the activity time and required workers remain the same. Effectiveness reduction probabilities and percentages were provided for each activity as subjective inputs by the ADTC point of contact. (See Appendix F Section 1)

Missions/Test Phase. Based on similarities in activity requirements and timing requirements, similar missions were then constructed and grouped into simulation test phases. Test phases are generally based on resource requirements and scheduling difficulties in order to collect statistics on these different areas of interest. Other criteria for

identifying test phases might include suspense dates for a particular group of missions, off-base testing, or different funding sources. (See Appendix F Section 2)

Specifically, the Pave Tack Night Attack Systems/F-4E test was divided into eight phases. The first two phases represent the integration tests which precede the DT&E testing. The last six phases represent the DT&E testing.

Phase 1 -- Integration Ground Missions. The first integration test phase represents the six ground missions. Two activities were used for each of these missions. The first activity (PMSNPG) included work associated with mission preparation. This activity is expected to require a substantial amount of preparation since the test engineers (TE) are just becoming familiar with the system. From 360 to 480 minutes were allowed for mission preparation. Three TE's are requested; however, test technicians (TT) or test analysts (TA) may be substituted if the TE's are not available. Also, the activity may be conducted with a partial team if three workers are not available. This activity was assigned a priority 2 since it must be completed the same day it is scheduled. The ground mission can then be conducted as scheduled on the following day.

The second activity (CGDMSN) represents the ground checkout procedures such as sensor alignment and power-on checks of all other equipment. The same number and order of personnel preference is required for both activities of a mission in this phase. An activity time between 120 and 180 minutes was selected for these checkout procedures.

Phase 2 -- Integration Air Missions. The second integration test phase represents the 12 flight test missions. Three activities were used

for each of these missions. The first activity (PMSNI) included the work associated with mission preparation. Since the crews are still in the learning process, an activity time between 480 and 600 minutes is required. Two TE's are involved in this familiarization process, but the activity may be conducted by only one worker.

The second activity (CMSNI) represents flight maneuvers and actions to be performed in checking out the system. An activity time between 120 and 180 minutes is required for two TE's. This activity has the potential of reducing the mission effectiveness; therefore, mission reduction probabilities and factors were entered.

The third activity (PSTMNI) represents a debriefing which follows each flight. A TT joins the two TE's to increase his knowledge about the system. A moderate amount of time was allowed (120 to 180 minutes) because of the lengthy discussions required during this learning period. The three cancellation probabilities entered for this test phase represent aircraft maintenance, test item maintenance, and weather.

Phase 3 -- DT&E Task A Missions. Following the integration phases and prior to the formal start of DT&E, test phase 3 represents the missions required for aircrew checkout and system optimization. The activities required for a mission in this test phase are similar to test phase 2 since there is no data reduction or analysis involved. The first activity (PMSNPA) represents mission preparation. The required activity time (90 to 120 minutes) is less than the previous test phases because crews are gaining proficiency with the equipment. This activity is primarily performed by TE's; however, it may also be performed by either TT's or TA's and with partial teams.

The second activity, conducting an air mission (CAMSN), also requires less time (105 to 135 minutes) than in previous phases because of the learning curve. This activity may only be accomplished by TE's because of the flying requirement. Mission effectiveness reduction probabilities and factors are high because of the increased emphasis on the system and the time frame of the test phase.

The third activity (PSTMNA) requires the same amount of time for debriefing (120 - 180 minutes); however, a TT or a TA is required, in addition to the TE's to assist in taking notes and viewing parts of the video tapes. An analysis activity is not required since the spot checks are only used to verify that the systems are functioning properly. The same mission cancellation factors as were used in test phase 2 are appropriate for missions in this test phase.

Phase 4 -- DT&E Objectives A1 through B2. Test phase 4 represents the missions required to determine system performance and operator capability with vairous target temperature differentials, weather, illumination, and background conditions. This phase is the largest and most time consuming phase of the test. It requires 22 missions consisting of seven activities each. The first three activities (PMSNPA, CAMSN, PSTMNA) are similar to the activities in test phase 3 except that the time and personnel requirements are increased because all missions consist of two aircraft sorties. The two sorties in each mission are flown during the same day.

The next two activities (ANTAPE, ANLOGS) represent the work necessary to analyze the video tapes from the aircraft and to review the mission logs. This analysis is normally performed by TA's after sortie

completion, but overtime is not required. The analysis may also be performed by TE's, TT's, and partial teams. A small reduction of mission effectiveness probability and factors was entered to represent problems discovered on the tapes.

The sixth activity (DATARD) represents the data reduction performed for each mission by the Computer Sciences Laboratory. The manpower requirements are external to the test force and thus zero was entered for the number of required workers. The activity priority is 1; and the activity time, consequently, represents the length of time before the data will be returned to the JTF. A small reduction of mission effectiveness probability and factors was entered to represent problems discovered during data reduction.

The last activity (DATAAN) can be accomplished after the reduced data is received from the Computer Sciences Laboratory. The activity time is extensive (1440-2160 minutes) since the work involves many hand calculations and graph plottings. The activity is normally performed by TA's but may be performed by TT's, TE's and partial teams. Mission cancellation probabilities were doubled based on the number of sorties.

An external mission was used in this test phase to represent a mission which had slightly different requirements. An additional analysis of tape activity (ANTAPE) was added which required more personnel than the initial tape analysis for the mission. All other activities in the mission remained the same.

Phase 5 -- DT&E Objective B3 through B7. The missions in test phase 5 are similar to test phase 4 missions except fewer test resources are required. Seven missions, each consisting of seven activities, are

required. The first five activities (PMSNPA, CAMSN, PSTMNA, ANTAPE, ANLOGS) are the same as the first five activities of test phase 4 missions except fewer workers are required. This requirement is a result of a reduction in the number of data collection sights.

The last two activities (DATARD, DATAAN) have a lower activity time because there is less data to reduce and analyze. The cancellation probability due to test item malfunction was reduced in this test phase to reflect a stabilizing trend in equipment performance.

Phase 6 -- DT&E Objective B8. Test phase 6 represents the missions conducted by JTF personnel while TDY to Nellis AFB. Two TE's were TDY for a period of 6 days. A priority 1 activity (MSNNEL) is used to represent these missions so that these two workers will not be released until the mission has been completed. This action insures that the simulation model does not permit these workers to be assigned to teams at Eglin AFB. The overtime statistic for these workers must be mentally adjusted to compensate for their off-duty hours.

An external event was used in this test phase to represent data analysis at Eglin AFB on the missions conducted at Nellis AFB. The Nellis AFB activity (MSNNEL) was deleted and an analysis activity (DATAAN) was added with its required entries. Any defined activity may be added to or deleted from a mission in a test phase. It is illustrative to note that this activity was performed in zero time because the arguments for the activity in the data input were not updated after adding the external event.

Phase 7 -- DT&E Objectives G through T and X and AA. The missions in this test phase represent several individual test objectives which have

common resource requirements. This subjective grouping was based on information presented in the required resource table and the accompanying text. Four missions were required consisting of 5 activities each. The first three activities (PMSNPA, CAMSN, PSTMNA) are similar to the first three activities in test phases 4 and 5, except only one aircraft sortie is required for each mission. The activity times and personnel requirements were reduced accordingly.

The last two activities (DATARD, DATAAN) were judged more complicated than the corresponding activity in test phase 5 because the data had to be cross-correlated between several sources. The number of required workers and the activity times were increased to accommodate this additional workload.

Phase 8 -- DT&E Objective W. Test phase 8 represents the missions required to demonstrate the weapon delivery accuracy of the Pave Tack weapon system. Ten missions, consisting of five activities each, were required. The same basic activities (PMSNPA, CAMSN, PSTMNA, DATARD, DATAAN) are required for a mission in test phase 8 as were used in test phase 7. The required work for four of these activities was judged to be less complicated than the corresponding activities in the test phase 7 missions; so the worker requirements and activity times were reduced accordingly. The data reduction activity (DATARD) required a longer activity time because the Computer Sciences Laboratory must hand process the aircraft position information.
(Ref 2)

Workers. After the activities are grouped into missions, the worker information must be determined. Nine workers were divided into three

levels for this sample run. Different skill factors were entered for the workers. This action reflects that workers were more proficient in performing activities which they normally performed by that worker level than for those activities which they only occasionally perform. For example, it would take a TT twice as long to perform a data analysis activity (ANLOGS) as it would take a TA to perform that same activity. (See Appendix F Section 3)

Program Variables. The program variables were entered as directed in the Data Input Instructions listed in Appendix D.

Activity Test Phase Codes. The two segments of the activity test phase codes were entered to correspond to the missions in each test phase. The first segment corresponds to the test phase number (1 through 8) based on the chronological order in which the missions were scheduled. The second segment corresponds to the order in which activities are to be performed and is the same order in which they were discussed in the respective test phase. All activity codes meet the requirement of having the same number of digits.

Mission Priorities. Priorities between missions of different test phases were explicitly assigned by prefixing the activity test phase codes. In the Pave Tack simulation test phase 5, missions were given a priority over all other missions by prefixing test phase 5 activity codes with a 1 and all other activity codes with a 2.

Options. Most of the options were illustrated in this example. Since only selected output from the final report could be reasonably presented in this thesis, weekly and monthly reports were not requested. Both reduction of mission effectiveness and mission cancellation were

entered. A workweek of 5 days was entered, and priority activities were not permitted to be worked on the weekends. Individual skill factors, as discussed in Workers, were entered for each worker. Special missions were scheduled using external event notices, and one run of the data package was requested.

Discussion of Output

Section 4. Selected outputs are presented in Appendix F Sections 4, 5, and 6 to show some of the results obtained by entering the sample data. The source of this data was discussed in the previous section. Section 4 shows the first and last test phase mission statistics. Since no cancellation factors were entered and no missions were less than 100% effective in test phase 1, 6 missions were scheduled and the effective productivity was also 6. Cancellation factors for test phase 8 were entered as reflected by the cancellation of 2 missions. Of the remaining 11 missions, the effective productivity was slightly more than 10 missions. Similar results were observed in the other test phases.

Section 5. Section 5 presents the statistics collected on the missions which were waiting for worker assignments. The state percentages show that most of the time the missions were being worked as quickly as they arrived. (Almost 60% of the time 2 or less missions were in the queue.) The percentages then peak at 7 - 8 missions indicating a backlog for a substantial period of time. Monthly reports could be consulted to show the phase and time frame of the backlog.

Section 6. Section 6 shows the information collected on the workers by levels and as individuals. The maximum number of idle workers in each level is 3, which corresponds to the total number of workers in the input

data. The total idle time is lengthy for all levels, and level 2 workers were used the least. The individual worker statistics indicate that TE2 and PO were sent TDY to Nellis AFB since their overtime hours are considerably larger than the other level 1 worker statistics. If cost data on overtime hours were to be determined, these figures would be adjusted to reflect the actual number of work hours per day.

Parametric Studies. Multiple runs were made with reduced personnel. Based on the results shown in the tables, the number of workers in levels 2 and 3 were reduced until a total of six workers were in the simulation. Fewer than six workers would have resulted in a programming fatal error since an activity in test phase 4 requires six workers. The completion date remained the same for all runs; and, as expected, the idle time in all worker levels progressively decreased as the number of workers decreased. For this model configuration a substantial savings can be realized without a change in completion date by reducing the personnel requirement by one-third.

IV. Conclusions and Extension of Research

The main thrust of this research project was the development of the event-oriented simulation model which is useful in estimating work completion dates and personnel requirements. The results obtained by entering the data for Pave Tack compared favorably with the ongoing test; however, as with all stochastic simulations, the results produced from a single run are of little significance. The illustrative example merely shows how to apply the simulation to a JTF and exercises many of the model's significant features. The real value of the simulation model will be realized through extensive parametric studies with replications in order to develop confidence intervals for the results.

The statistics accumulated in the simulation model readily lend themselves to cost-benefit analysis for various work rules versus completion dates. The flexibility of data input allows the model to accurately simulate situations which have yet to be considered. A continuation of this study should concentrate on the application of the simulation model to extensively test its intended flexibility. Techniques for establishing confidence intervals for the output data should be developed. In conjunction with the confidence intervals, different seed values for the random number generators should be tested for significance.

Further research should determine which distributions can be used to better describe the activity times. Uniform distributions were used with parameter estimates verified by JTF members in order to insure structural validity of the simulation model. Since the simulation results compare favorably with the actual test, more descriptive distribu-

tions can now be selected to refine the estimates. Guidance on which distributions should be used with various classes of activities would be of value to the Test Wing.

Additional research could be conducted in the areas of mission cancellations and reductions in mission effectiveness. Extensive data is available in this area, but time permitted only a superficial analysis of this data to provide inputs for the sample run. This information, which is accumulated by the Test Wing, should be further studied to provide better estimates under a variety of conditions. Additional analysis should compare this past data with the final statistics of Pave Tack. some differences can be expected; because, under the JTF concept, some resources are dedicated to the test force. If there is an appreciable difference, future data inputs should favor the results obtained under the JTF concept.

Bibliography

1. Air Force Systems Command. RF-4C/Flir Reconnaissance System. Eglin AFB, Florida, 1976.

2. Air Force Systems Command. F-4E/Night Attack System. Eglin AFB, Florida, 1976.

3. Armament Development and Test Center. Test Engineers' Handbook. Eglin AFB, Florida, 1976.

4. Armament Development and Test Center. Data Reduction and Computational Capabilities. Eglin AFB, Florida, 1975.

5. Emshoff, James R. and Roger L. Sission. Design and Use of Computer Simulation Models. New York: Macmillan Publishing Co., Inc., 1970.

6. Fishman, George S. and Phillip J. Kiviat, Digital Computer Simulation: Statistical Considerations, (AD66 0953; California: The Rand Corporation, November 1967).

7. Fryer, J. S. "Operating Policies in Multiechelon Dual-constraint Job Shops." Management Science, 19:1001-1012 (May 1973).

8. Hillier, F. S. and G. J. Lieberman. Introduction to Operations Research (Second Edition). San Francisco, Calif.: Holden-Day Inc., 1974.

9. Johnson, Glen and Edward C. Russell. SIMSCRIPT II.5 Users Manual, Honeywell 600.6000 Version. Los Angeles, Calif.: CACI, Inc. 1973.

10. Kiviat, Phillip J., Computer Simulation Programming Languages: Perspective and Prognosis, (AD659358; California: The Rand Corporation, September 1967).

11. Kiviat, Phillip J., Digital Computer Simulation: Computer Programming Languages (AD84124; California: The Rand Corporation, January 1969).

12. Kiviat, P. J., R. Villanueva, and H. M. Markowitz, SIMSCRIPT II.5 Programming Language. Los Angeles, Calif.: CACI, Inc. 1973

13. Naylor, T. H. and J. M. Finger. "Verification of Computer Simulation Models." Management Science, 14:B92-B106 (October 1967).

14. Russel, Edward C. Simulation and SIMSCRIPT II.5 (Preliminary Printing). Los Angeles, Calif: CACI, Inc., 1975.

15. Samuelson, Paul A. Economics (Eighth Edition). New York: McGraw-Hill Book Co., 1970.

16. Schribner, Thomas J. Simulation Using GPSS. New York: John Wiley and Sons, 1974.

17. Tocher, K. D. "Review of Simulation Languages." Operations Research, 16:189-217 (June 1965).

18. Van Horn, R. L. "Validation of Simulation Results." Management Science, 17:247-258 (January 1971).

APPENDIX A

DEFINITION OF TERMS

ATTRIBUTES OF TEMPORARY ENTITY "ACTIVITY"

ACEL - CURRENT EFFECTIVENESS LEVEL
EACH ACTIVITY HAS THE POTENTIAL TO REDUCE THE EFFECTIVENESS OF THE MISSION. THE ACEL SHOWS THE MISSION EFFECTIVENESS AT THE END OF THE ACTIVITY.

AFWL - FIRST WORKER LEVEL
FIRST OF THRFE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND.OF.ACTIVITY.

AICT - INTERRUPTION CONTINUATION TIME
TIME REQUIRED TO COMPLETE AN ACTIVITY WHICH HAS BEEN EITHER PREEMPTED OR INTERRUPTED

ALRL - LOWER REDUCTION LIMIT
LOWER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THAT THE ACEL IS TO BE REDUCED.

ANOA - NUMBER OF ACTIVITY
SAME NUMBER AS THE CORRESPONDING KIND.OF.ACTIVITY

ANWR - NUMBER OF WORKERS REQUIRED
EACH ACTIVITY MAY REQUIRE FROM ZERO TO THE TOTAL NUMBER OF WORKERS IN THE SIMULATION TO PERFORM THE ACTIVITY.

APAR - PARTIAL TEAM CAPABILITY
YES/NO ATTRIBUTE WHICH INDICATES WHETHER AN ACTIVITY CAN BE PERFORMED WITH FEWER WORKERS THAN THE NUMBER REQUESTED.

APMA - PERCENT MISSIONS AFFECTED
IF A RANDOM NUMBER IS LESS THAN OR EQUAL TO THIS VALUE, THE ACEL WILL BE REDUCED BY AN AMOUNT BETWEEN THE ALRL AND THE AURL.

APOA - PRIORITY OF ACTIVITY
ONE OF THREE PRIORITIES WHICH INDICATES THE TIME OF DAY THAT AN ACTIVITY CAN BE PERFORMED AND ITS ABILITY TO TAKE WORKERS AWAY FROM LOWER PRIORITY MISSIONS.

ASWL - SECOND WORKER LEVEL
SECOND OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND.OF.ACTIVITY.

ATFC - TIME FUNCTION CODE
INDICATES THE TYPE OF DISTRIBUTION FROM WHICH AN ACTIVITY TIME WILL BE SELECTED.

ATPC - TEST PHASE CODE
SEQUENTIAL CODE NUMBERS ASSIGNED TO ACTIVITIES WHICH COMPRISE A MISSION IN EACH TEST PHASE. PRIORITIES MAY BE ASSIGNED BETWEEN THE DIFFERENT TEST PHASES BY PREFIXING THE CODE WITH PRIORITY NUMBERS. (LOW NUMBER IS HIGH PRIORITY)

ATWL - THIRD WORKER LEVEL
THIRD OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND.OF.ACTIVITY

AURL - UPPER REDUCTION LIMIT
UPPER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THE ACEL IS TO BE REDUCED.

ATTRIBUTES OF EVENT NOTICES

CTPN - CONFIRMED MISSION TEST PHASE NUMBER
PROGRAMMING VALUE USED TO DETERMINE THE TEST PHASE OF THE NEXT MISSION.

EAMN - END OF ACTIVITY MISSION NUMBER
PROGRAMMING VALUE TO IDENTIFY THE MISSION NUMBER OF A COMPLETED ACTIVITY.

ATTRIBUTES OF PERMANENT ENTITY "KIND.OF.ACTIVITY"

KPAR - PARTIAL TEAM CAPABILITY
PERMANENT COPY OF YES/NO ATRRIBUTE OF EACH KIND OF ACTIVITY WHICH INDECATES WHETHER AN ACTIVITY CAN BE PERFORMED WITH FEWER WORKERS THAN THE NUMBER REQUESTED.

KFWL - FIRST WORKER LEVEL
PERMANENT COPY OF THE FIRST OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.

KLRL - LOWER REDUCTION LIMIT
PERMANENT COPY OF LOWER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THAT THE ACEL IS TO BE REDUCED.

KPMA - PERCENT MISSIONS AFFECTED
PERMANENT COPY OF THE MAXIMUM VALUE OF A RANDOM NUMBER WHICH INDICATES THAT THE MISSION EFFECTIVENESS IS TO BE REDUCED BY AN AMOUNT BETWEEN KLRL AND KURL.

KPOA - PRIORITY OF ACTIVITY
PERMANENT COPY DESIGNATING ONE OF THE THREE PRIORITIES WHICH INDICATES THE TIME OF DAY THAT AN ACTIVITY CAN BE PERFORMED AND ITS ABILITY TO TAKE WORKERS AWAY FROM LOWER PRIORITY MISSIONS.

KSWL - SECOND WORKER LEVEL
PERMANENT COPY OF THE SECOND OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.

KTFC - TIME FUNCTION CODE
PERMANENT COPY OF THE TYPE OF DISTRIBUTION FROM WHICH AN ACTIVITY TIME WILL BE SELECTED.

KTWL - THIRD WORKER LEVEL
PERMANENT COPY OF THE THIRD OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.

KURL - UPPER REDUCTION LIMIT
PERMANENT COPY OF THE UPPER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THAT THE ACEL IS TO BE REDUCED.

ATTRIBUTES OF PERMANENT ENTITY "LEVEL.OF.WORKER"

LNIW - NUMBER OF IDLE WORKERS
CONTAINS THE VALUES INDICATING THE NUMBER OF IDLE WORKERS IN EACH LEVEL. USED TO GATHER STATISTICS AND FOR PROGRAM CONTROL.

LTIT - TOTAL IDLE TIME
CONTAINS THE SUM OF THE IDLE TIME IN A LEVEL. ITS VALUES ARE PRINTED IN THE REPORTS.

ATTRIBUTES OF TEMPORARY ENTITY "MISSION"

MCEL - CURRENT EFFECTIVENESS LEVEL
VALUE OF MISSION EFFECTIVENESS AND MAY BE REDUCED BY EACH ACTIVITY IN THE MISSION.

MNAC - NEXT ACTIVITY CODE
NUMBER CORRESPONDING TO THE ATPC OF THE NEXT ACTIVITY OF THE MISSION. VALUE IS USED TO ORDER THE MISSIONS WAITING IN THE QUEUE.

MNAP - NEXT ACTIVITY PRIORITY
NUMBER CORRESPONDING TO THE APOA OF THE NEXT ACTIVITY OF THE MISSION. VALUE IS USED TO ORDER THE MISSIONS WAITING IN THE QUEUE.

MTOA - TIME OF ARRIVAL
TIME AT WHICH NOTIFICATION OF A MISSION WAS RECEIVED. VALUE IS USED TO DETERMINE MISSION DURATION AND THE TIME AT WHICH A PRIORITY 1 MISSION CAN BE CONDUCTED.

MTPN - TEST PHASE NUMBER
IDENTIFIES THE TEST PHASE TO WHICH THE MISSION BELONGS

MXCI - CANCELL CODE INDICATOR
IF THE VALUE IS 1 WHEN THE MISSION RECORD RETURNS FROM DETERMINE.PREMISSION.CANCELLATION OR CALCULATE. MISSION.EFFECTIVENESS, THE MISSION IS CANCELLED AND 1 IS ADDED TO THE NUMBER OF CANCELLED MISSIONS IN THE TEST PHASE.

ATTRIBUTES OF PERMANENT ENTITY "TEST.PHASE"

TCMP - NUMBER OF COMPLETED MISSIONS
CURRENT COUNT IN EACH TEST PHASE

TCNF - NUMBER OF CONFIRMED MISSIONS
CURRENT COUNT IN EACH TEST PHASE. SOME OF THESE MISSIONS MAY BE CANCELLED.

TCNX - NUMBER OF CANCELLED MISSIONS
CURRENT COUNT IN EACH TEST PHASE.

TDBM - DAYS BEFORE NEXT MISSION
IF THE VALUE IS GREATER THAN OR EQUAL TO 1 IT IS USED WITH THE TTMC RANDOM VARIABLE TO DETERHINE THE TIME OF THE NEXT MISSION. IF THE VALUE IS LESS THAN 1, IT IS ADDED TO THE CURRENT SIMULATION TIME. IF THE SUM IS DURING NORMAL DUTY HOURS, THE MISSION IS SCHEDULED AT THAT TIME; OTHERWISE, THE MISSION IS SCHEDULED FOR

THE BEGINNING OF DUTY HOURS ON THE NEXT DUTY DAY.

TDCD - DATE COMPLETED
DATE THAT THE LAST MISSION WAS COMPLETED IN EACH TEST PHASE.

TNUM - NUMBER OF TEST PHASE
NUMBERS RANGE FROM 1 TO THE TOTAL NUMBER OF TEST PHASES

TSTY - STAY TIME STATISTIC
VALUE COMPUTED FOR EACH MISSION IN A TEST PHASE AS THE DIFFERENCE BETWEEN THE TIME COMPLETED AND THE TIME OF NOTIFICATION.

TTMC - TIME MISSION CONFIRMED RANDOM VARIABLE
USER PROVIDED DISTRIBUTION FOR THE TIME OF DAY THAT THE MISSION IS TO BE SCHEDULED.

TREQ - NUMBER OF REQUIRED MISSIONS
VALUE TO WHICH THE SUM OF ALL FINAL MCEL MUST EQUAL OR EXCEED IN EACH TEST PHASE.

GLOBAL VARIABLES

VBB1T - BEGIN BREAK 1 TIME

VBB2T - BEGIN BREAK 2 TIME

VBDDT - BEGIN DUTY DAY TIME

VBLBT - BEGIN LUNCH BREAK TIME

VEB1T - END BREAK 1 TIME

VEB2T - END BREAK 2 TIME

VEDDT - END DUTY DAY TIME

VELBT - END LUNCH BREAK TIME

VMROP - MONTHLY REPORT OPTION
ENTERING NO IN THE INPUT DATA DELETES THIS REPORT.

VRMPT - REQ MISSION PREPARATION TIME (INCLUDED IN DURATION TIME)
TIME REQUIRED TO PREPARE FOR AN ACTIVITY. (BETWEEN MISSION NOTIFICATION AND THE FIRST PRIORITY 1 ACTIVITY) IF ALL PREMISSION ACTIVITIES HAVE BEEN COMPLETED PRIOR TO THIS TIME, THE MISSION IS FILED IN WAIT.FOR.MISSION.TIME.

VRUNS - NUMBER OF SIMULATION RUNS FOR ENTIRE DATA PACKAGE
NUMBER OF TIMES THE ENTIRE DATA PACKAGE IS TO BE RUN WITH DIFFERENT SEED VALUES IN THE RANDOM NUMBER GENERATORS TO PROVIDE A DISTRIBUTION OF VALUES.

VWEAM - WEEKEND ACTIVE MISSIONS
YES OR NO OPTION TO ALLOW PRIORITY MISSIONS TO BE CONDUCTED ON WEEKENDS.

VWROP - WEEKLY REPORT OPTION
ENTERING NO IN THE INPUT DATA DELETES THIS REPORT

VWWDD - WORK WEEK DUTY DAYS
NUMBER OF DAYS IN THE NORMAL DUTY WEEK. TO INTERNALLY SCHEDULE MISSIONS ON THE WEEKEND THIS VALUE MUST BE INCREASED ACCORDINGLY.

REPORT VARIABLES

VWCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST WEEK

VWCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST WEEK
VWCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST WEEK
VMCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST MONTH
VMCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST MONTH
VMCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST MONTH

LOCAL VARIABLES(WITH A SINGLE LETTER PREFIX OR SUFFIX)
"AZ" - ACTIVITY....EXAMPLE AZG OR PAZ
"MZ" - MISSION.....EXAMPLE MZF OR QMZ
"TZ" - TEST PHASE..EXAMPLE TZB OR YTZ
"WZ" - WORKER.....EXAMPLE WZD OR UWZ

ATTRIBUTES OF PERMANENT ENTITY "WORKER"
WLEV - LEVEL
CATAGORY OR TYPE OF WORKER
WNAM - NAME
NAME OF EACH WORKER (6 DIGETS OR LESS).
WSTA - STATE
SHOWS THE CURRENT STATUS OF THE WORKER AS IDLE, BUSY, BREAK, OFF DUTY, OR OVERTIME.

ATTRIBUTES OF TEMPORARY ENTITY "XFACTOR"
XUPL - UPPER PERCENTAGE LIMIT
IF A RANDOM NUMBER IS LESS THAN OR EQUAL TO THIS VALUE THE MISSION IS CANCELLED.

PROGRAMMING CODES

ACTIVITY PRIORITY CODES

1 - ACTIVE MISSION, PREEMPT WORKERS, CONTINUE UNTIL FINISHED
2 - MUST BE COMPLETED IN CURRENT DAY
3 - NORMAL DUTY HOURS

WORKER STATUS CODES

0 - IDLE
1 - BUSY
2 - BREAK
3 - OFF.DUTY
4 - OVERTIME

TIME FUNCTION CODES

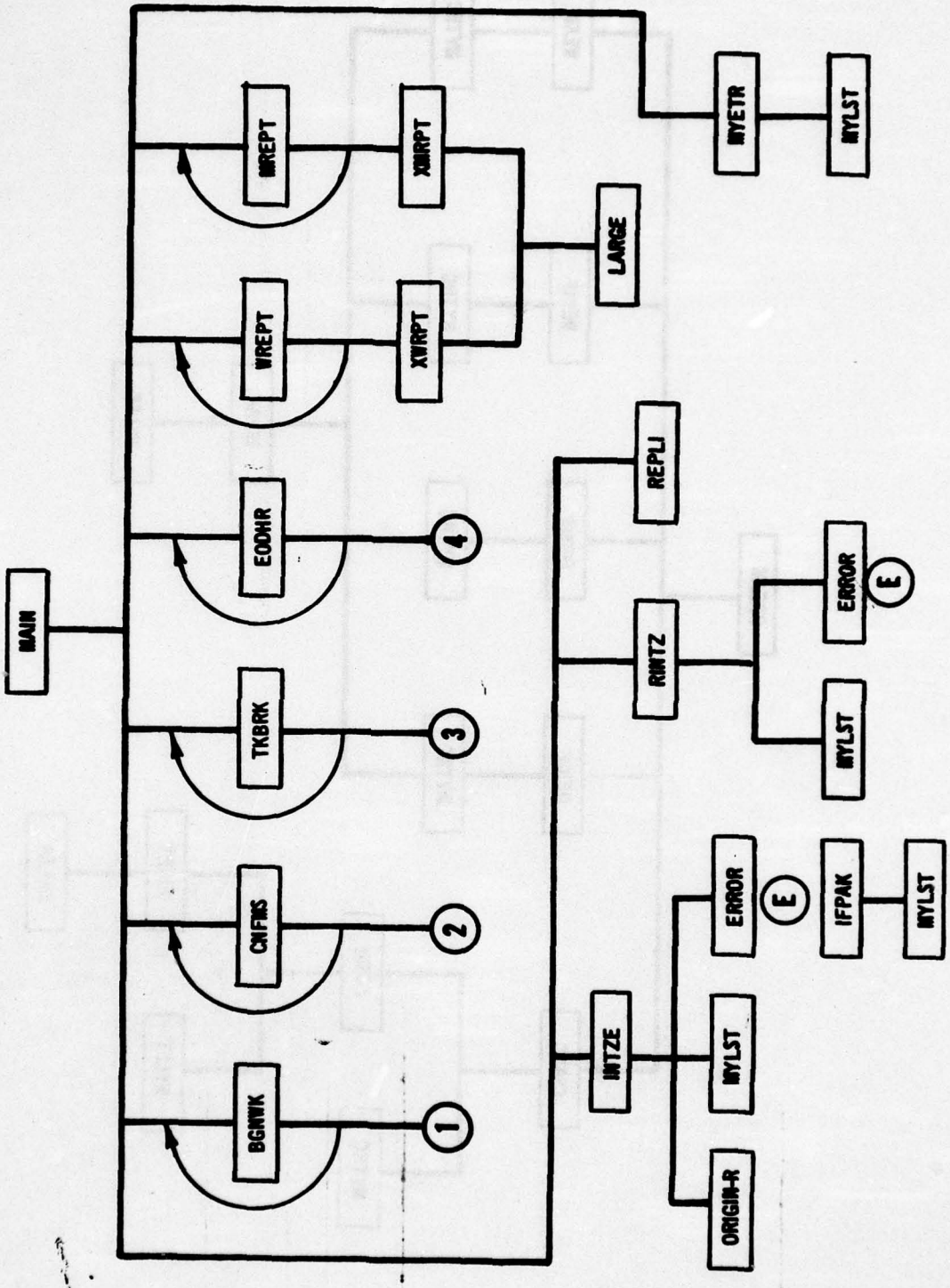
		ARGUMENTS	
		1ARG	2ARG
1	BETA	POWER OF X	POWER OF (1-X)
2	BINOMIAL	NUMBER OF TRIALS	PROB OF SUCCESS
3	CONSTANT	MEAN	VARIANCE
4	ERLANG	MEAN	SHAPE PARAMETER
5	EXPONENTIAL	MEAN	MEAN
6	GAMMA	MEAN	SHAPE PARAMETER
7	NORMAL	MEAN	STD DEVIATION
8	POISSON	MEAN	MEAN
9	UNIFORM	BEGINNING VALUE	ENDING VALUE
10	WEIBULL	SCALE PARAMETER	SHAPE PARAMETER
-1	INTEGER STEP	N/A	N/A
-2	REAL STEP	N/A	N/A
-3	CONTINUOUS	N/A	N/A

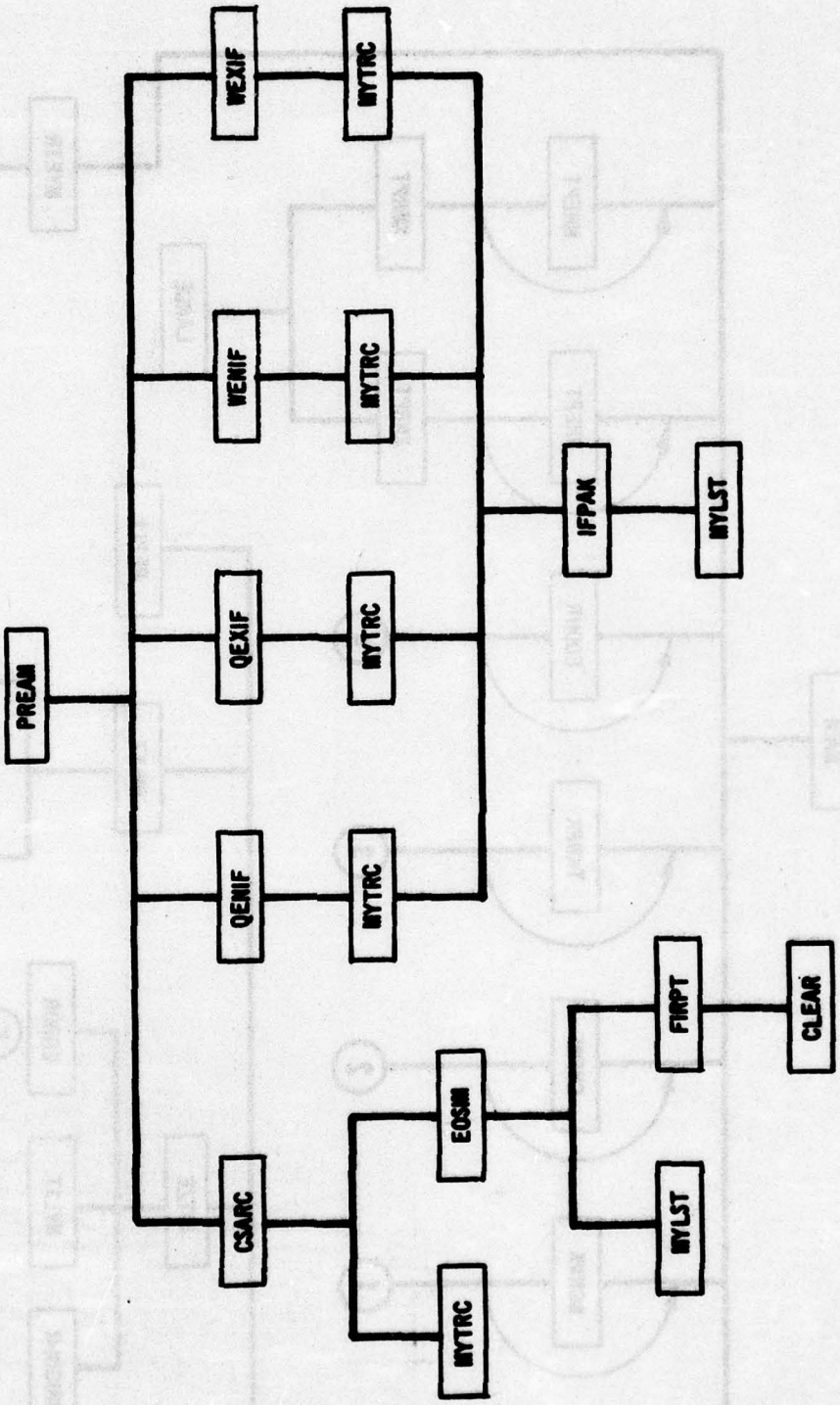
APPENDIX B

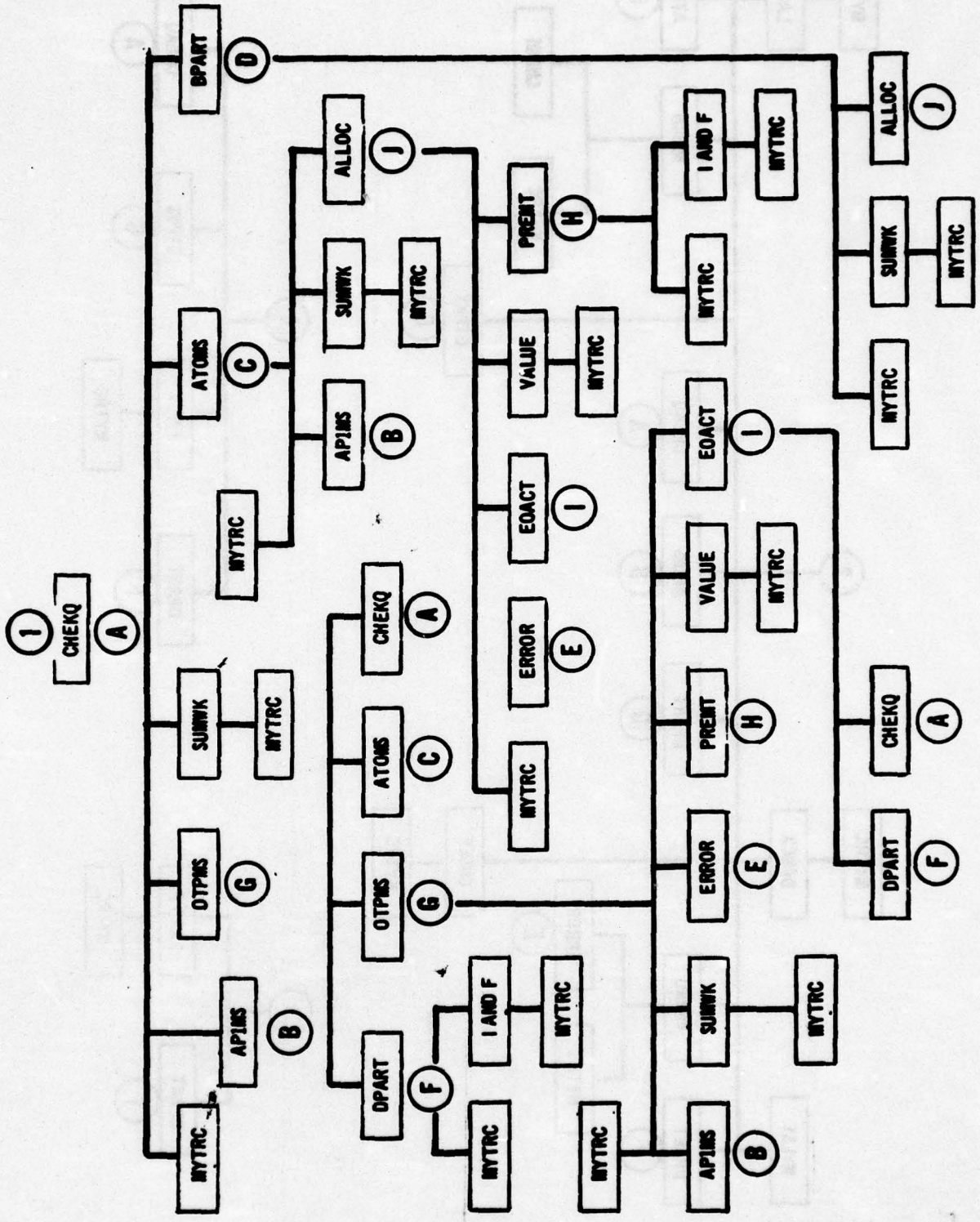
FLOW CHART

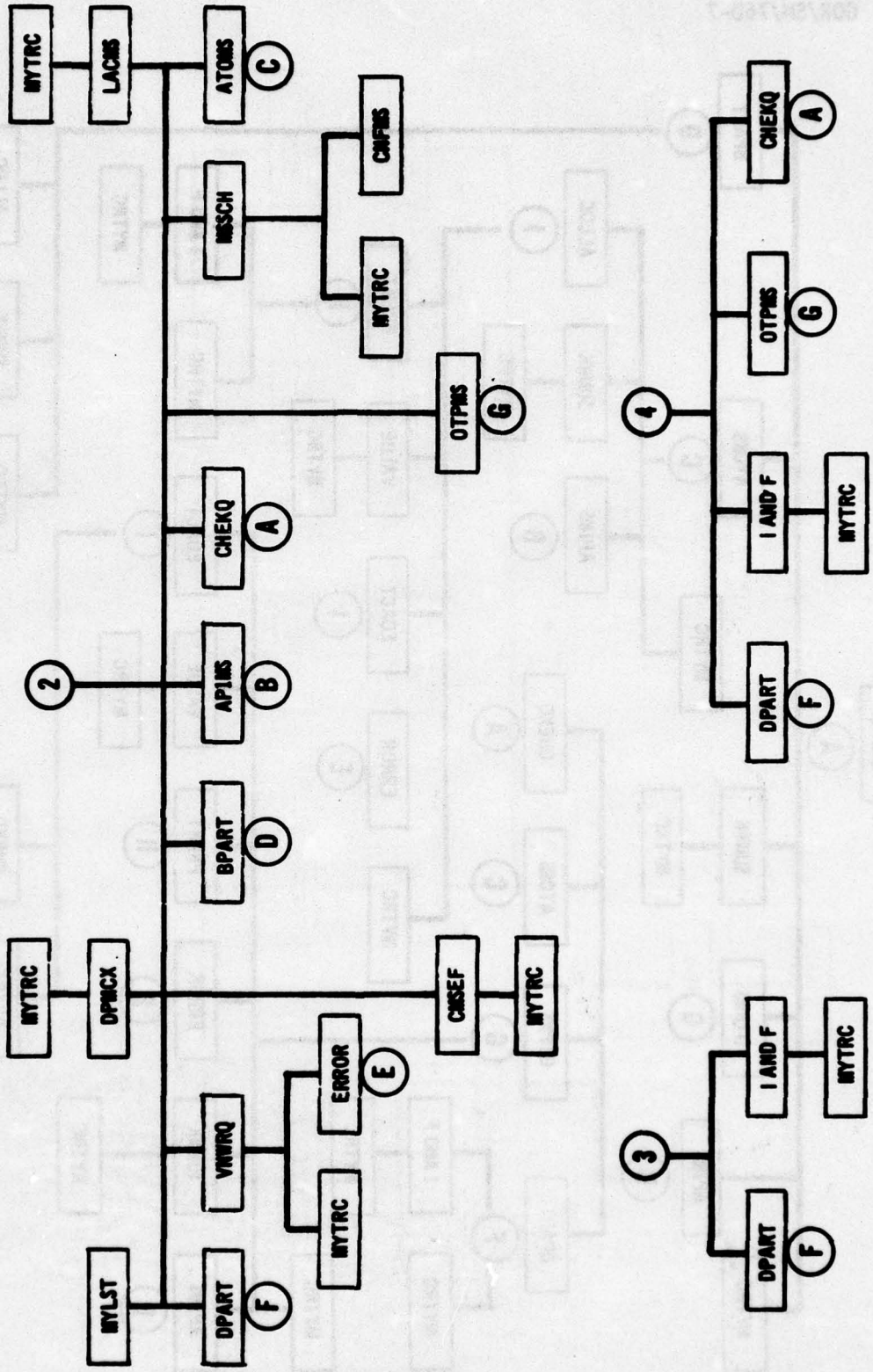
Flow Chart Abbreviations

ABBREVIATION	EVENT OR ROUTINE
ALLOC	ALLOCATION
APlMS	ACCOMPLISH.PRl.MISSION
ATOMS	ATTEND.TO.MISSION
BGNWK	BEGIN.WORKING
BPART	BUILD.PARTIAL.TEAMS
CHEKQ	CHECK.QUEUE
CLEAR	CLEAR
CMSEF	CALCULATE.MISSION.EFFECTIVENESS
CNFMS	CONFIRMED.MISSION
CSARC	COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION
DPART	DISBAND.PARTIAL.TEAMS
DPMCX	DETERMINE.PREMISSION.CANCELLATION
EOACT	EO.ACTIVITY
EODHR	EO.DUTY.HOURS
EOSIM	EO.SIMULATION
ERROR	ERROR.PACKAGE
FIRPT	FINAL.REPORT
IANDF	INTERRUPT.AND.FILE
IFPAK	INFO.PACKAGE
INTZE	INITIALIZE
LACMS	LOOK.AHEAD.COMPLETE.MISSIONS
LARGE	LARGEST
MAIN	MAIN
MREPT	MREPORT
MSSCH	MISSION.SCHEDULE
MYETR	MYE.TRACE
MYLST	MY.LIST
MYTRC	MY.TRACE
OTPMS	OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
PREAM	PREAMBLE
PREMT	PREEMPT
QENIF	QENTER.INFO
QEXIF	QEXIT.INFO
REPLI	REPLICATION
RINTZ	RUN.INITIALIZATION
SUMWK	SUM.WORKERS
TKBRK	TAKE.BREAK
VALUE	VALUE
VNWRQ	VALIDATE.NUMBER.OF.WORKERS.REQUESTED
WENIF	WENTER.INFO
WEXIF	WENTER.INFO
WREPT	WREPORT
XMRPT	XMREPORT
XMRPT	XWREPORT


MAIN
BGNWK
CNFMS
TKBRK
EODHR
WREPT
XWRPT
MREPT
XMRPT
LARGE
MYETR
MYLST
1
2
3
4
REPLI
RWNTZ
ERROR
E
MYLST
WNTZE
MYLST
ERROR
E
ORIGIN-R
IFPAK
MYLST


PREAM
WEXIF
MYTRC
WENIF
MYTRC
QEXIF
MYTRC
QENIF
MYTRC
IFPAK
MYLST
CSARC
EOSIM
MYTRC
FIRPT
CLEAR
MYLST


1
CHEKQ
A
MYTRC
APIMS
B
OTPMS
G
SUMWK
MYTRC
ATOMS
C
MYTRC
APIMS
B
SUMWK
MYTRC
ALLOC
J
DPART
D
CHEKQ
A
DPART
F
I AND F
MYTRC
OTPMS
G
ATOMS
C
MYTRC
EOACT
I
ERROR
E
MYTRC
VALUE
MYTRC
PRENT
H
MYTRC
APIMS
B
SUMWK
MYTRC
ERROR
E
PRENT
H
VALUE
MYTRC
EOACT
I
MYTRC
I AND F
MYTRC
DPART
F
CHEKQ
A
MYTRC
SUMWK
MYTRC
ALLOC
J


MYTRC
LACMS
ATOMS
C
MSSCH
CMFMS
MYTRC
OTPMS
G
CHIEKQ
A
2
APIMS
B
BPART
D
MYTRC
DPMCX
CMSEF
MYTRC
MYLST
DPART
F
VMMRQ
MYTRC
ERROR
E
4
CHIEKQ
A
OTPMS
G
I AND F
MYTRC
DPART
F
3
I AND F
MYTRC
DPART
F

APPENDIX C

PROGRAM LISTING

```
**                    SIMSCRIPT II.5 SIMULATION PROGRAM                    00000010
**                       JOINT TEST FORCE SIMULATION                       00000020
**                                                                         00000030
                                    PREAMBLE**                             00000040
  LAST COLUMN IS 72**                                                      00000050
  NORMALLY MODE IS INTEGER AND DIMENSION IS 0                              00000060
**                                                                         00000070
**DECLARATION OF STATIC SYSTEM STRUCTURE                                   00000080
**                                                                         00000090
  PERMANENT ENTITIES...                                                    00000100
    EVERY TEST.PHASE HAS A TTMC RANDOM LINEAR VARIABLE, A TNUM, A TREQ,    00000110
      A TCNF, A TCMP, A TCNX, A TDCD, A TDBM, A TSTY, OWNS A STRUCTURE,    00000120
      AND OWNS A TCANCELLATION.PROBABILITY                                 00000130
      DEFINE TTMC AS A REAL VARIABLE                                       00000140
    EVERY KIND.OF.ACTIVITY HAS A KNAM, A KPOA, A KTFC, A KPAR, A KFWL,     00000150
      A KSWL, A KTWL, A KPMA, A KLRL, A KURL                               00000160
    EVERY LEVEL.OF.WORKER HAS A LNIW, A LTIT                               00000170
    EVERY WORKER HAS A WNAM, A WLEV, A WSTA, AND MAY BELONG TO A TEAM      00000180
    EVERY TEST.PHASE,KIND.OF.ACTIVITY HAS A 1ARG AND A 2ARG                00000190
      DEFINE 1ARG AND 2ARG AS REAL VARIABLES                               00000200
    EVERY WORKER,KIND.OF.ACTIVITY HAS A SKILL.FACTOR                       00000210
      DEFINE SKILL.FACTOR AS A REAL VARIABLE                               00000220
**EMPERICAL FUNCTIONS                                                      00000230
    EVERY INTEGER.STEP.FUNCTION HAS A ONE RANDOM STEP VARIABLE             00000240
       DEFINE ONE AS AN INTEGER STREAM 9 VARIABLE                          00000250
    EVERY REEL.STEP.FUNCTION HAS A TWO RANDOM STEP VARIABLE                00000260
      DEFINE TWO AS A REAL STREAM 9 VARIABLE                               00000270
    EVERY CONTINUOUS.FUNCTION HAS A THREE RANDOM LINEAR VARIABLE           00000280
      DEFINE THREE AS A REAL STREAM 9 VARIABLE                             00000290
**                                                                         00000300
  TEMPORARY ENTITIES...                                                    00000310
    EVERY MISSION HAS A MTPN, A MCEL, A MTOA, A MNAC, A MNAP, A MXCI,      00000320
        MAY BELONG TO THE QUEUE, THE WAIT.FOR.MISSION.TIME,                00000330
     OWNS A ROUTING, AND A MCANCELLATION.PROBABILITY                       00000340
    EVERY ACTIVITY HAS A ATPC, A ANOA, A APOA, A ATFC, A AICT, A ANWR,     00000350
```

```
      A APAR, A AFWL, A ASWL, A ATWL, A ACEL, A APMA, A ALRL, A AURL      00000360
      BELONGS TO A STRUCTURE AND A ROUTING, AND OWNS A TEAM               00000370
    EVERY XFACTOR HAS A XUPL, AND BELONG TO A                             00000380
      TCANCELLATION.PROBABILITY AND A MCANCELLATION.PROBABILITY           00000390
**                                                                        00000400
  THE SYSTEM OWNS THE QUEUE                                             00000410
    DEFINE QUEUE AS A FIFO SET RANKED BY LOW MNAP, THEN BY LOW MNAC,    00000420
      THEN BY LOW MTOA                                                  00000430
  THE SYSTEM OWNS THE WAIT.FOR.MISSION.TIME                             00000440
    DEFINE WAIT.FOR.MISSION.TIME AS A FIFO SET RANKED BY LOW MTOA       00000450
**                                                                        00000460
**DECLARATION OF DYNAMIC SYSTEM STRUCTURE                                00000470
**                                                                        00000480
  EVENT NOTICES INCLUDE BEGIN.WORKING, TAKE.BREAK, EO.DUTY.HOURS,       00000490
    ACCOMPLISH.PR1.MISSION, WREPORT, MREPORT,                           00000500
     EO.SIMULATION, AND FINAL.REPORT                                    00000510
    EVERY CONFIRMED.MISSION HAS A CTPN                                  00000520
    EVERY EO.ACTIVITY HAS A EAMN                                        00000530
  EXTERNAL EVENTS ARE CONFIRMED.MISSION AND EO.SIMULATION               00000540
  EXTERNAL EVENT UNITS ARE 8 AND 9                                      00000550
  PRIORITY ORDER IS EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,          00000560
    CONFIRMED.MISSION, TAKE.BREAK, ACCOMPLISH.PR1.MISSION,              00000570
    BEGIN.WORKING, WREPORT, MREPORT, AND FINAL.REPORT                   00000580
  DEFINE CSTARC TO MEAN COMPUTE.STAY.TIME.AND.RECORD.CANCELLATIONS      00000590
  BEFORE DESTROYING MISSION CALL CSTARC                                 00000600
  BEFORE FILING IN QUEUE CALL QENTER.INFO                               00000610
  BEFORE FILING IN WAIT.FOR.MISSION.TIME CALL WENTER.INFO               00000620
  BEFORE REMOVING FROM QUEUE CALL QEXIT.INFO                            00000630
  BEFORE REMOVING FROM WAIT.FOR.MISSION.TIME CALL WEXIT.INFO            00000640
**                                                                        00000650
**DATA COLLECTION AND ANALYSIS DECLARATIONS                              00000660
**                                                                        00000670
    NORMALLY MODE IS REAL                                               00000680
  TALLY WAVG.STAY AS THE WEEKLY MEAN, WSTD.STAY AS THE WEEKLY STD.DEV,  00000690
    WSUM.STAY AS THE WEEKLY SUM, WSOME.SQUARES.STAY AS THE WEEKLY       00000700
```

```
      SUM.OF.SQUARES, AND WNUM.STAY AS THE WEEKLY NUMBER OF TSTY          00000710
    TALLY MAVG.STAY AS THE MONTHLY MEAN,MSTD.STAY AS THE MONTHLY STD.DEV, 00000720
      MSUM.STAY AS THE MONTHLY SUM, MSOME.SQUARES.STAY AS THE MONTHLY     00000730
      SUM.OF.SQUARES, AND MNUM.STAY AS THE MONTHLY NUMBER OF TSTY         00000740
    TALLY TAVG.STAY AS THE MEAN, TSTD.STAY AS THE STD.DEV, TSUM.STAY AS   00000750
      THE SUM, TSOME.SQUARES.STAY AS THE SUM.OF.SQUARES, AND TNUM.STAY AS 00000760
      THE NUMBER OF TSTY                                                  00000770
    ACCUMULATE WQSUM.QUEUE AS THE WEEKLY SUM, WQSTD.QUEUE AS THE WEEKLY   00000780
      STD.DEV,  WQAVG.QUEUE AS THE WEEKLY MEAN, WQMIN.QUEUE AS THE WEEKLY 00000790
      MINIMUM, WQMAX.QUEUE AS THE WEEKLY MAXIMUM AND WFREQ(0 TO 500 BY 1) 00000800
      AS THE WEEKLY HISTOGRAM OF N.QUEUE                                  00000810
    ACCUMULATE MQSUM.QUEUE AS THE MONTHLY SUM, MQSTD.QUEUE AS THE MONTHLY 00000820
      STD.DEV, MQAVG.QUEUE AS THE MONTHLY MEAN, MQMIN.QUEUE AS THE MONTHLY00000830
      MINIMUM, MQMAX.QUEUE AS THE MONTHLY MAXIMUM AND MFREQ(0 TO 500 BY 1)00000840
      AS THE MONTHLY HISTOGRAM OF N.QUEUE                                 00000850
    ACCUMULATE TQSUM.QUEUE AS THE SUM, TQSTD.QUEUE AS THE STD.DEV,        00000860
       TQAVG.QUEUE AS THE MEAN, TQMAX.QUEUE AS                            00000870
      THE MAXIMUM, AND TFREQ(0 TO 500 BY 1) AS THE HISTOGRAM OF N.QUEUE   00000880
    ACCUMULATE WLMAX.IDLE.WORKERS AS THE WEEKLY MAXIMUM,                  00000890
               MLMAX.IDLE.WORKERS AS THE MONTHLY MAXIMUM,                 00000900
               TLMAX.IDLE.WORKERS AS THE MAXIMUM,                         00000910
               WIDLE.WORKERS(0 TO 100 BY 1) AS THE WEEKLY HISTOGRAM,      00000920
               MIDLE.WORKERS(0 TO 100 BY 1) AS THE MONTHLY HISTOGRAM, AND 00000930
               TIDLE.WORKERS(0 TO 100 BY 1) AS THE HISTOGRAM OF LNIW      00000940
    ACCUMULATE WOBSERVED.STATE(0 TO 4 BY 1) AS THE WEEKLY HISTOGRAM,      00000950
               MOBSERVED.STATE(0 TO 4 BY 1) AS THE MONTHLY HISTOGRAM, AND 00000960
               TOBSERVED.STATE(0 TO 4 BY 1) AS THE HISTOGRAM OF WSTA      00000970
      NORMALLY MODE IS INTEGER                                            00000980
**                                                                        00000990
**MISC DECLARATIONS                                                      00001000
**                                                                        00001010
      DEFINE IDLE TO MEAN 0                                               00001020
      DEFINE BUSY TO MEAN 1                                               00001030
      DEFINE BREAK TO MEAN 2                                              00001040
      DEFINE OFF.DUTY TO MEAN 3                                           00001050
```

```
   DEFINE OVERTIME TO MEAN 4                                                    00001060
   DEFINE WK TO MEAN *HOURS.V*7 HOURS                                           00001070
   DEFINE NEXT.MONTH TO                                                         00001080
     MEAN AT REAL.F(DATE.F(MONTH.F(TIME.V+31),1,YEAR.F(TIME.V+31)))             00001090
   DEFINE KOA TO MEAN KIND.OF.ACTIVITY                                          00001100
   DEFINE LOFW TO MEAN LEVEL.OF.WORKER                                          00001110
   DEFINE N.LOFW TO MEAN N.LEVEL.OF.WORKER                                      00001120
   DEFINE SAME.SKILL.FACTOR, DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS,         00001130
   CHECK, EMPERICAL.FUNCTIONS, APAR, KNAM, KPAR, VMROP, VWEAM, VWROP,          00001140
     AND WNAM AS ALPHA VARIABLES                                                00001150
   DEFINE ACEL, AICT, ALRL, APMA, AURL, KLRL, KPMA, KURL, LTIT, MCEL,          00001160
     MTOA, TCMP, TDBM, TDCD, TSTY, VBB1T, VBB2T, VBDDT, VBLBT, VEB1T,           00001170
     VEB2T, VEDDT, VELBT, VRMPT, XUPL, CK.MIN, CK.MAX AS REAL VARIABLES         00001180
   DEFINE VRUNS, MONTH, DAY, YEAR, VWWDD AS INTEGER VARIABLES                   00001190
   DEFINE SAVESEED, VWCNF, VWCNX, VMCNF, VMCNX AS                               00001200
     1-DIMENSIONAL INTEGER VARIABLES                                            00001210
   DEFINE VWCMP, VMCMP AS 1-DIMENSIONAL REAL VARIABLES                          00001220
   DEFINE SUM.WORKERS AS AN INTEGER FUNCTION WITH 3 ARGUMENTS                   00001230
   DEFINE LACM TO MEAN LOOK.AHEAD.COMPLETE.MISSIONS                             00001240
   DEFINE LACM AS A REAL FUNCTION WITH 1 ARGUMENT                               00001250
   DEFINE LARGEST AS A REAL FUNCTION WITH 1 ARGUMENT                            00001260
   DEFINE VALUE AS A REAL FUNCTION WITH 1 ARGUMENT                              00001270
   DEFINE INITIALIZE, REPLICATION, XWREPORT, XMREPORT AS                        00001280
     RELEASABLE ROUTINES                                                        00001290
END                                                                             00001300
```

```
**                                                                    00001310
                                        MAIN                        00001320
  *REPLICATE*                                                       00001330
  CALL INITIALIZE                                                   00001340
  IF VRUNS = 1                                                      00001350
    RELEASE INITIALIZE                                              00001360
    RELEASE REPLICATION                                             00001370
  ALWAYS                                                            00001380
  LET BETWEEN.V = *MYE.TRACE*                                       00001390
  *SIMULATE*                                                        00001400
  CALL RUN.INITIALIZATION                                           00001410
  SCHEDULE A BEGIN.WORKING AT VBDDT/24                              00001420
  SCHEDULE A CONFIRMED.MISSION(1) AT 8/24                           00001430
  SCHEDULE A TAKE.BREAK AT VBB1T/24                                 00001440
  SCHEDULE A BEGIN.WORKING AT VEB1T/24                              00001450
  SCHEDULE A TAKE.BREAK AT VBLBT/24                                 00001460
  SCHEDULE A BEGIN.WORKING AT VELBT/24                              00001470
  SCHEDULE A TAKE.BREAK AT VBB2T/24                                 00001480
  SCHEDULE A BEGIN.WORKING AT VEB2T/24                              00001490
  SCHEDULE A EO.DUTY.HOURS AT VEDDT/24                              00001500
  IF VWROP = "NO"                                                   00001510
    RELEASE XWREPORT                                                00001520
    GO TO MRPT                                                      00001530
  ELSE                                                              00001540
  SCHEDULE A WREPORT IN 1 WK                                        00001550
  *MRPT*                                                            00001560
  IF VMROP = "NO"                                                   00001570
    RELEASE XMREPORT                                                00001580
    GO TO START                                                     00001590
  ELSE                                                              00001600
  SCHEDULE A MREPORT NEXT.MONTH                                     00001610
  *START*                                                           00001620
  START SIMULATION                                                  00001630
  USE UNIT 7 FOR INPUT                                              00001640
  IF DATA IS ENDED                                                  00001650
```

```
    IF VRUNS GT 1                                                          00001660
      CALL REPLICATION                                                     00001670
      GO REPLICATE                                                         00001680
    ELSE                                                                   00001690
    STOP                                                                   00001700
  ELSE                                                                     00001710
**RUN SIMULATION WITH DIFFERENT DATA                                       00001720
  START NEW PAGE                                                           00001730
  PRINT 1 LINE THUS                                                        00001740
  RUN SIMULATION WITH DIFFERENT WORKERS AND TIME BETWEEN MISSIONS          00001750
  GO *SIMULATE*                                                            00001760
  END                                                                      00001770
```

```
**                                                                    00001780
                          ROUTINE TO INITIALIZE                       00001790
USE UNIT 7 FOR INPUT                                                  00001800
IF VRUNS NE 0                                                         00001810
  SKIP 4 FIELDS                                                       00001820
  GO READ                                                             00001830
ELSE                                                                  00001840
READ VRUNS, MONTH, DAY, YEAR                                          00001850
CALL ORIGIN.R(MONTH,DAY,YEAR)                                         00001860
*READ*                                                                00001870
READ VWROP, VMROP,                                                    00001880
  N.KIND.OF.ACTIVITY, DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS       00001890
  CREATE EVERY KIND.OF.ACTIVITY                                       00001900
  IF DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS = "YES"                00001910
    FOR EACH KIND.OF.ACTIVITY DO                                      00001920
      READ KNAM(KOA), KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA),     00001930
        KSWL(KOA), KTWL(KOA)                                          00001940
      LET KPMA(KOA) = 0          **IF AN ACTIVITY CANNOT BE PERFORMED 00001950
      LET KLRL(KOA) = 0          **BY 3 DIFFERENT LEVELS OF WORKER    00001960
      LET KURL(KOA) = 0          **MAKE REPEATED ENTRIES OF A QUALIFIED00001970
    LOOP                         **LEVEL OF WORKER                    00001980
    JUMP AHEAD                                                        00001990
  ELSE                                                                00002000
  FOR EACH KIND.OF.ACTIVITY                                           00002010
    READ KNAM(KOA), KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA),       00002020
    KSWL(KOA), KTWL(KOA), KPMA(KOA), KLRL(KOA), KURL(KOA)             00002030
    HERE                                                              00002040
READ N.TEST.PHASE                                                     00002050
  CREATE EVERY TEST.PHASE                                             00002060
  FOR EACH TEST.PHASE DO                                              00002070
    READ TNUM(TEST.PHASE), TREQ(TEST.PHASE), TTMC(TEST.PHASE)         00002080
    FOR EACH KIND.OF.ACTIVITY                                         00002090
      READ 1ARG(TEST.PHASE,KOA), 2ARG(TEST.PHASE,KOA)                 00002100
    UNTIL MODE IS ALPHA DO                                            00002110
      CREATE AN ACTIVITY CALLED BAZ                                   00002120
```

```
      READ ATPC(BAZ), ANOA(BAZ), ANWR(BAZ)                          00002130
      LET APOA(BAZ) = KPOA(ANOA(BAZ))                               00002140
      LET ATFC(BAZ) = KTFC(ANOA(BAZ))                               00002150
      LET AICT(BAZ) = 0                                             00002160
      LET APAR(BAZ) = KPAR(ANOA(BAZ))                               00002170
      LET AFWL(BAZ) = KFWL(ANOA(BAZ))                               00002180
      LET ASWL(BAZ) = KSWL(ANOA(BAZ))                               00002190
      LET ATWL(BAZ) = KTWL(ANOA(BAZ))                               00002200
      LET ACEL(BAZ) = 1                                             00002210
      LET APMA(BAZ) = KPMA(ANOA(BAZ))                               00002220
      LET ALRL(BAZ) = KLRL(ANOA(BAZ))                               00002230
      LET AURL(BAZ) = KURL(ANOA(BAZ))                               00002240
      FILE BAZ IN STRUCTURE(TEST.PHASE)                             00002250
    LOOP                                                            00002260
    SKIP 1 FIELD                                                    00002270
    **TO DELETE THIS FEATURE ENTER A "*"                            00002280
    UNTIL MODE IS ALPHA DO                                          00002290
      CREATE AN XFACTOR                                             00002300
      READ XUPL                                                     00002310
      FILE XFACTOR IN TCANCELLATION.PROBABILITY                     00002320
    LOOP                                                            00002330
    SKIP 1 FIELD                                                    00002340
  LOOP                                                              00002350
READ CHECK                                                          00002360
IF CHECK IS "OK"                                                    00002370
  JUMP AHEAD                                                        00002380
ELSE                                                                00002390
READ EMPERICAL.FUNCTIONS                                            00002400
IF EMPERICAL.FUNCTIONS = "NO"                                       00002410
  GO TO NONE                                                        00002420
ELSE                                                                00002430
READ N.INTEGER.STEP.FUNCTION                                        00002440
  CREATE EVERY INTEGER.STEP.FUNCTION                                00002450
  FOR EVERY INTEGER.STEP.FUNCTION                                   00002460
    READ ONE(INTEGER.STEP.FUNCTION)                                 00002470
```

```
READ N.REEL.STEP.FUNCTION                                              00002480
  CREATE EVERY REEL.STEP.FUNCTION                                      00002490
  FOR EVERY REEL.STEP.FUNCTION                                         00002500
    READ TWO(REEL.STEP.FUNCTION)                                       00002510
READ N.CONTINUOUS.FUNCTION                                             00002520
  CREATE EVERY CONTINUOUS.FUNCTION                                     00002530
  FOR EVERY CONTINUOUS.FUNCTION                                        00002540
    READ THREE(CONTINUOUS.FUNCTION)                                    00002550
  "NONE"                                                               00002560
READ VWWDD, VWEAM, VBDDT, VBB1T, VEB1T, VBLBT, VELBT,                  00002570
     VBB2T, VEB2T, VEDDT, VRMPT                                        00002580
READ CHECK                                                             00002590
IF CHECK NE "OK"                                                       00002600
  JUMP AHEAD                                                           00002610
ELSE                                                                   00002620
RESERVE SAVESEED AS 10                                                 00002630
FOR I = 1 TO 10                                                        00002640
  LET SAVESEED(I) = SEED.V(I)                                          00002650
START NEW PAGE                                                         00002660
PRINT 131 LINES THUS                                                   00002670
                        ############################                   00002680
                        ############################                   00002690
                        JOINT TEST FORCE SIMULATION                    00002700
                                                                       00002710
                          ABBREVIATIONS AND CODES                      00002720
                        ############################                   00002730
                        ############################                   00002740
                                                                       00002750
                                                                       00002760
      ATTRIBUTES OF TEMPORARY ENTITY "ACTIVITY"                        00002770
ACEL - CURRENT EFFECTIVENESS LEVEL                                     00002780
AFWL - FIRST WORKER LEVEL                                              00002790
AICT - INTERRUPTION CONTINUATION TIME                                  00002800
ALRL - LOWER REDUCTION LIMIT                                           00002810
ANOA - NUMBER OF ACTIVITY                                              00002820
```

```
ANWR - NUMBER OF WORKERS REQUIRED                                  00002830
APAR - PARTIAL TEAM CAPABILITY                                     00002840
APMA - PERCENT MISSIONS AFFECTED                                   00002850
APOA - PRIORITY OF ACTIVITY                                        00002860
ASWL - SECOND WORKER LEVEL                                         00002870
ATFC - TIME FUNCTION CODE                                          00002880
ATPC - TEST PHASE CODE                                             00002890
ATWL - THIRD WORKER LEVEL                                          00002900
AURL - UPPER REDUCTION LIMIT                                       00002910
                                                                   00002920
     ATTRIBUTES OF EVENT NOTICES                                   00002930
CTPN - CONFIRMED MISSION TEST PHASE NUMBER                         00002940
EAMN - END OF ACTIVITY MISSION NUMBER                              00002950
                                                                   00002960
     ATTRIBUTES OF PERMANENT ENTITY "KIND.OF.ACTIVITY"             00002970
KFWL - FIRST WORKER LEVEL                                          00002980
KLRL - LOWER REDUCTION LIMIT                                       00002990
KPMA - PERCENT MISSIONS AFFECTED                                   00003000
KPAR - PARTIAL TEAM CAPABILITY                                     00003010
KPOA - PRIORITY OF ACTIVITY                                        00003020
KSWL - SECOND WORKER LEVEL                                         00003030
KTFC - TIME FUNCTION CODE                                          00003040
KTWL - THIRD WORKER LEVEL                                          00003050
KURL - UPPER REDUCTION LIMIT                                       00003060
                                                                   00003070
     ATTRIBUTES OF PERMANENT ENTITY "LEVEL.OF.WORKER"              00003080
LNIW - NUMBER OF IDLE WORKERS                                      00003090
LTIT - TOTAL IDLE TIME                                             00003100
                                                                   00003110
     ATTRIBUTES OF TEMPORARY ENTITY "MISSION"                      00003120
MCEL - CURRENT EFFECTIVENESS LEVEL                                 00003130
MNAC - NEXT ACTIVITY CODE                                          00003140
MNAP - NEXT ACTIVITY PRIORITY                                      00003150
MTOA - TIME OF ARRIVAL                                             00003160
MTPN - TEST PHASE NUMBER                                           00003170
```

```
MXCI - CANCEL CODE INDICATOR                                                00003180
                                                                            00003190
       ATTRIBUTES OF PERMANENT ENTITY "TEST.PHASE"                          00003200
TCMP - NUMBER OF COMPLETED MISSIONS                                         00003210
TCNF - NUMBER OF CONFIRMED MISSIONS                                         00003220
TCNX - NUMBER OF CANCELLED MISSIONS                                         00003230
TDBM - DAYS BEFORE NEXT MISSION                                             00003240
TDCD - DATE COMPLETED                                                       00003250
TNUM - NUMBER OF TEST PHASE                                                 00003260
TSTY - STAY TIME STATISTIC                                                  00003270
TTMC - TIME MISSION CONFIRMED RANDOM VARIABLE                               00003280
TREQ - NUMBER OF REQUIRED MISSIONS                                          00003290
                                                                            00003300
       GLOBAL VARIABLES                                                     00003310
VBB1T - BEGIN BREAK 1 TIME                                                  00003320
VBB2T - BEGIN BREAK 2 TIME                                                  00003330
VBDDT - BEGIN DUTY DAY TIME                                                 00003340
VBLBT - BEGIN LUNCH BREAK TIME                                              00003350
VEB1T - END BREAK 1 TIME                                                    00003360
VEB2T - END BREAK 2 TIME                                                    00003370
VEDDT - END DUTY DAY TIME                                                   00003380
VELBT - END LUNCH BREAK TIME                                                00003390
VMROP - MONTHLY REPORT OPTION                                               00003400
VRMPT - REQ MISSION PREPARATION TIME (INCLUDED IN DURATION TIME)             00003410
VRUNS - NUMBER OF SIMULATION RUNS FOR ENTIRE DATA PACKAGE                   00003420
VWEAM - WEEKEND ACTIVE MISSIONS                                             00003430
VWROP - WEEKLY REPORT OPTION                                                00003440
VWWDD - WORK WEEK DUTY DAYS                                                 00003450
                                                                            00003460
       REPORT VARIABLES                                                     00003470
VWCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST WEEK             00003480
VWCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST WEEK             00003490
VWCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST WEEK             00003500
VMCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST MONTH            00003510
VMCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST MONTH            00003520
```

```
VMCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST MONTH      00003530
                                                                      00003540
        LOCAL VARIABLES(WITH A SINGLE LETTER PREFIX OR SUFFIX)        00003550
"AZ" - ACTIVITY....EXAMPLE  AZG OR PAZ                                00003560
"MZ" - MISSION.....EXAMPLE  MZF OR QMZ                                00003570
"TZ" - TEST PHASE..EXAMPLE  TZB OR YTZ                                00003580
"WZ" - WORKER......EXAMPLE  WZD OR UWZ                                00003590
                                                                      00003600
        ATTRIBUTES OF PERMANENT ENTITY "WORKER"                       00003610
WLEV - LEVEL                                                          00003620
WNAM - NAME                                                           00003630
WSTA - STATE                                                          00003640
                                                                      00003650
        ATTRIBUTES OF TEMPORARY ENTITY "XFACTOR"                      00003660
XUPL - UPPER PERCENTAGE LIMIT                                         00003670
                                                                      00003680
                              PROGRAMMING CODES                       00003690
                                                                      00003700
        ACTIVITY PRIORITY CODES                                       00003710
1 - ACTIVE MISSION, PREEMPT WORKERS, CONTINUE UNTIL FINISHED          00003720
2 - MUST BE COMPLETED IN CURRENT DAY                                  00003730
3 - NORMAL DUTY HOURS                                                 00003740
                                                                      00003750
        WORKER STATUS CODES                                           00003760
0 - IDLE                                                              00003770
1 - BUSY                                                              00003780
2 - BREAK                                                             00003790
3 - OFF.DUTY                                                          00003800
4 - OVERTIME                                                          00003810
                                                                      00003820
                                                                      00003830
        TIME FUNCTION CODES                ARGUMENTS                  00003840
                                    1ARG                2ARG          00003850
1 - BETA                      POWER OF X          POWER OF (1-X)      00003860
2 - BINOMIAL----------NUMBER OF TRIALS-----PROBABILITY OF SUCCESS     00003870
```

```
 3 - CONSTANT                MEAN                    VARIANCE                00003880
 4 - ERLANG                  MEAN                    SHAPE PARAMETER         00003890
 5 - EXPONENTIAL-----------MEAN-------------------MEAN                       00003900
 6 - GAMMA                   MEAN                    SHAPE PARAMETER         00003910
 7 - NORMAL                  MEAN                    STANDARD DEVIATION      00003920
 8 - POISSON---------------MEAN-------------------MEAN                       00003930
 9 - UNIFORM                 BEGINNING VALUE         ENDING VALUE            00003940
10 - WEIBULL                 SCALE PARAMETER         SHAPE PARAMETER         00003950
-1 - INTEGER STEP----------N/A--------------------N/A                        00003960
-2 - REAL STEP              N/A                     N/A                     00003970
-3 - CONTINUOUS             N/A                     N/A                     00003980
START NEW PAGE                                                              00003990
PRINT 11 LINES THUS                                                         00004000
                                                                            00004010
                                                                            00004020
                    ########################################                00004030
                    INFORMATION COMMON TO ALL SIMULATION RUNS                00004040
                    ########################################                00004050
                                                                            00004060
                                                                            00004070
                                                                            00004080
                      ------------------------------------                  00004090
                      INFORMATION COMMON TO ALL TEST PHASES                 00004100
                      ------------------------------------                  00004110
CALL MY.LIST(12)**KIND OF ACTIVITY                                          00004120
CALL MY.LIST(9)**1ARG                                                       00004130
CALL MY.LIST(10)**2ARG                                                      00004140
SKIP 3 LINES                                                                00004150
PRINT 10 LINES WITH MONTH, DAY, YEAR, VWWDD, VBDDT, VBB1T, VEB1T,           00004160
  VBLBT, VELBT, VBB2T, VEB2T, VEDDT THUS                                    00004170
SIMULATION BEGINS---------------------- **,** ****                         00004180
THE NUMBER OF DUTY DAYS PER WEEK IS--   *                                   00004190
THE DUTY DAY BEGINS AT----------------- **.**                               00004200
THE MORNING COFFEE BREAK BEGINS AT---- **.**                                00004210
THE MORNING COFFEE BREAK ENDS AT------ **.**                                00004220
```

```
THE LUNCH PERIOD BEGINS AT---------- **.**                              00004230
THE LUNCH PERIOD ENDS AT------------ **.**                              00004240
THE AFTERNOON COFFEE BREAK BEGINS AT- **.**                             00004250
THE AFTERNOON COFFEE BREAK ENDS AT--- **.**                             00004260
THE DUTY DAY ENDS AT---------------- **.**                              00004270
PRINT 7 LINES THUS                                                      00004280
                                                                        00004290
                                                                        00004300
                                                                        00004310
                          ----------------------                        00004320
                          TEST PHASE INFORMATION                        00004330
                          ----------------------                        00004340
                                                                        00004350
FOR EACH TEST.PHASE DO                                                  00004360
  PRINT 2 LINES THUS                                                    00004370
##### NUMBER      MISSION TIME          ACTIVITY SEQUENCE AND RQMTS     00004380
                 PROB      TIME       CODE   ACTIVITY    WORKERS REQ   00004390
  LET I = F.TTMC(TEST.PHASE)                                            00004400
  LET J = F.STRUCTURE(TEST.PHASE)                                       00004410
  PRINT 1 LINE WITH TNUM(TEST.PHASE), PROB.A(I), RVALUE.A(I), ATPC(J),  00004420
  ANOA(J), ANWR(J) THUS                                                 00004430
       **        *.**    **.**         **         **         **        00004440
  UNTIL I=J=0 DO                                                        00004450
    IF I NE 0                                                           00004460
      LET I = S.TTMC(I)                                                 00004470
      THEN IF I NE 0                                                    00004480
        WRITE PROB.A(I) AND RVALUE.A(I) AS B 16,2 D(9,2)                00004490
    ALWAYS                                                              00004500
    IF J NE 0                                                           00004510
      LET J = S.STRUCTURE(J)                                            00004520
      THEN IF J NE 0                                                    00004530
        WRITE ATPC(J), ANOA(J), ANWR(J) AS B 35, 3 I 11                 00004540
    ALWAYS                                                              00004550
    START NEW LINE                                                      00004560
  LOOP                                                                  00004570
```

```
   CALL MY.LIST(3,TEST.PHASE)**TEST PHASE ATTRIBUTES                    00004580
   CALL MY.LIST(4,TEST.PHASE)**ACTIVITIES IN STRUCTURE(TEST PHASE)      00004590
   CALL MY.LIST(8,TEST.PHASE)**XUPL IN T.CX.PROB(TEST PHASE)            00004600
   SKIP 5 LINES                                                         00004610
LOOP                                                                    00004620
RETURN                                                                  00004630
HERE                                                                    00004640
CALL ERROR.PACKAGE(1)                                                   00004650
END                                                                     00004660
```

```
..
                              ROUTINE FOR RUN.INITIALIZATION                    00004670
  USE UNIT 7 FOR INPUT                                                          00004680
  UNTIL MODE IS ALPHA                                                           00004690
    READ CK.MIN, CK.MAX                                                         00004700
  SKIP 1 FIELD                                                                  00004710
  READ CHECK                                                                    00004720
  IF CHECK NE "OK"                                                              00004730
    JUMP AHEAD                                                                  00004740
  ELSE                                                                          00004750
  READ SAME.SKILL.FACTOR                                                        00004760
  READ N.LEVEL.OF.WORKER                                                        00004770
    CREATE EVERY LEVEL.OF.WORKER                                                00004780
  READ N.WORKER                                                                 00004790
    CREATE EVERY WORKER                                                         00004800
    FOR EACH WORKER DO                                                          00004810
      READ WNAM(WORKER), WLEV(WORKER)                                           00004820
      LET WSTA(WORKER) = OFF.DUTY                                               00004830
      IF SAME.SKILL.FACTOR = "YES"                                              00004840
        FOR EACH KIND.OF.ACTIVITY                                               00004850
          LET SKILL.FACTOR(WORKER,KOA) = 1                                      00004860
        GO LOOP                                                                 00004870
      ELSE                                                                      00004880
      FOR EACH KIND.OF.ACTIVITY                                                 00004890
        READ SKILL.FACTOR(WORKER,KOA)                                           00004900
        "LOOP"                                                                  00004910
    LOOP                                                                        00004920
    START NEW PAGE                                                              00004930
  PRINT 7 LINES THUS                                                            00004940
                                                                                00004950
                                                                                00004960
                                                                                00004970
                                                                                00004980
                ##################################################              00004990
                INFORMATION ENTERED FOR CURRENT SIMULATION RUN                  00005000
                ##################################################              00005010
```

```
PRINT 5 LINE THUS                                                    00005020
                                                                     00005030
                                                                     00005040
                                                                     00005050
                    ------------------------------------             00005060
                    INFORMATION COMMON TO ALL TEST PHASES            00005070
                    ------------------------------------             00005080
CALL MY.LIST(1)**WORKERS                                             00005090
CALL MY.LIST(11)**SKILL FACTOR                                       00005100
PRINT 8 LINES THUS                                                   00005110
                                                                     00005120
                                                                     00005130
                          ----------------------                     00005140
                          TEST PHASE INFORMATION                     00005150
                          ----------------------                     00005160
                                                                     00005170
                                                                     00005180
TEST PHASE     DAYS BETWEEN MISSIONS                                 00005190
FOR EACH TEST.PHASE                                                  00005200
  READ TDBM(TEST.PHASE)                                              00005210
FOR EVERY TEST.PHASE PRINT 1 LINE WITH TEST.PHASE AND                00005220
  TDBM(TEST.PHASE) THUS                                              00005230
    ***              **.**                                           00005240
PRINT 8 LINES THUS                                                   00005250
                                                                     00005260
                                                                     00005270
                                                                     00005280
                          ####################                       00005290
                          ####################                       00005300
                          END OF INPUT SUMMARY                       00005310
                          ####################                       00005320
                          ####################                       00005330
SKIP 10 LINES                                                        00005340
READ CHECK                                                           00005350
HERE                                                                 00005360
```

```
  IF CHECK NE "OK"                                          00005370
    CALL ERROR.PACKAGE(1)                                   00005380
  ALWAYS                                                    00005390
RETURN                                                      00005400
END                                                         00005410
```

```
**                                                                    00005580
                          ROUTINE TO CHECK.QUEUE                      00005590
CALL MY.TRACE(1)                                                      00005600
DEFINE T AS A REAL VARIABLE                                           00005610
FOR EVERY MZJ IN QUEUE WITH MNAP(MZJ) = 1 AND                         00005620
  MTOA(MZJ) + VRMPT GT TIME.V DO                                      00005630
  REMOVE MZJ FROM QUEUE                                               00005640
  FILE MZJ IN WAIT.FOR.MISSION.TIME                                   00005650
  SCHEDULE AN ACCOMPLISH.PR1.MISSION AT MTOA(MZJ) + VRMPT             00005660
LOOP                                                                  00005670
IF FRAC.F(TIME.V) GE VEDDT/24 OR FRAC.F(TIME.V) LT VBDDT/24           00005680
  GO TO OVERTIME                                                      00005690
ELSE                                                                  00005700
IF WEEKDAY.F(TIME.V) GT VWWDD                                         00005710
  *OVERTIME*                                                          00005720
  CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                      00005730
  RETURN                                                              00005740
ELSE                                                                  00005750
**IF CHECKED DURING BREAK, PR1 MISSIONS SHOULD BE INITIATED           00005760
LET T = FRAC.F(TIME.V) * 24                                           00005770
IF VBB1T LT T LE VEB1T OR                                             00005780
   VBLBT LT T LE VELBT OR                                             00005790
   VBB2T LT T LE VEB2T                                                00005800
  FOR EVERY MZJ IN QUEUE WITH MNAP(MZJ) = 1 AND                       00005810
    ANWR(F.ROUTING(MZJ)) LE SUM.WORKERS(2,MZJ,0) DO                   00005820
    REMOVE MZJ FROM QUEUE                                             00005830
    CALL MY.TRACE(2,MZJ)                                              00005840
    CALL ATTEND.TO.MISSION GIVING MZJ                                 00005850
  LOOP                                                                00005860
ALWAYS                                                                00005870
FOR EVERY DMZ IN QUEUE WITH ANWR(F.ROUTING(DMZ)) LE                   00005880
  SUM.WORKERS(1,DMZ,0) DO                                             00005890
  REMOVE DMZ FROM QUEUE                                               00005900
  CALL ATTEND.TO.MISSION GIVING DMZ                                   00005910
LOOP                                                                  00005920
```

```
**                                                                  00005420
                EVENT BEGIN.WORKING SAVING THE EVENT NOTICE         00005430
SCHEDULE THIS BEGIN.WORKING IN 24 HOURS                             00005440
IF WEEKDAY.F(TIME.V) GT VWWDD                                       00005450
  RETURN **OFFICE IS CLOSED FOR THE WEEKEND                         00005460
ELSE                                                                00005470
FOR EVERY WORKER CALLED WZU WITH WSTA(WZU) = BREAK OR                00005480
                        WSTA(WZU) = OFF.DUTY DO                     00005490
  LET WSTA(WZU) = IDLE                                              00005500
  ADD 1 TO LNIW(WLEV(WZU))                                          00005510
LOOP                                                                00005520
FOR EVERY WORKER CALLED WZV WITH WSTA(WZV) = OVERTIME               00005530
  LET WSTA(WZV) = BUSY                                              00005540
CALL CHECK.QUEUE                                                    00005550
RETURN                                                              00005560
END                                                                 00005570
```

```
CALL BUILD.PARTIAL.TEAMS                                        00005930
RETURN                                                          00005940
END                                                             00005950
```

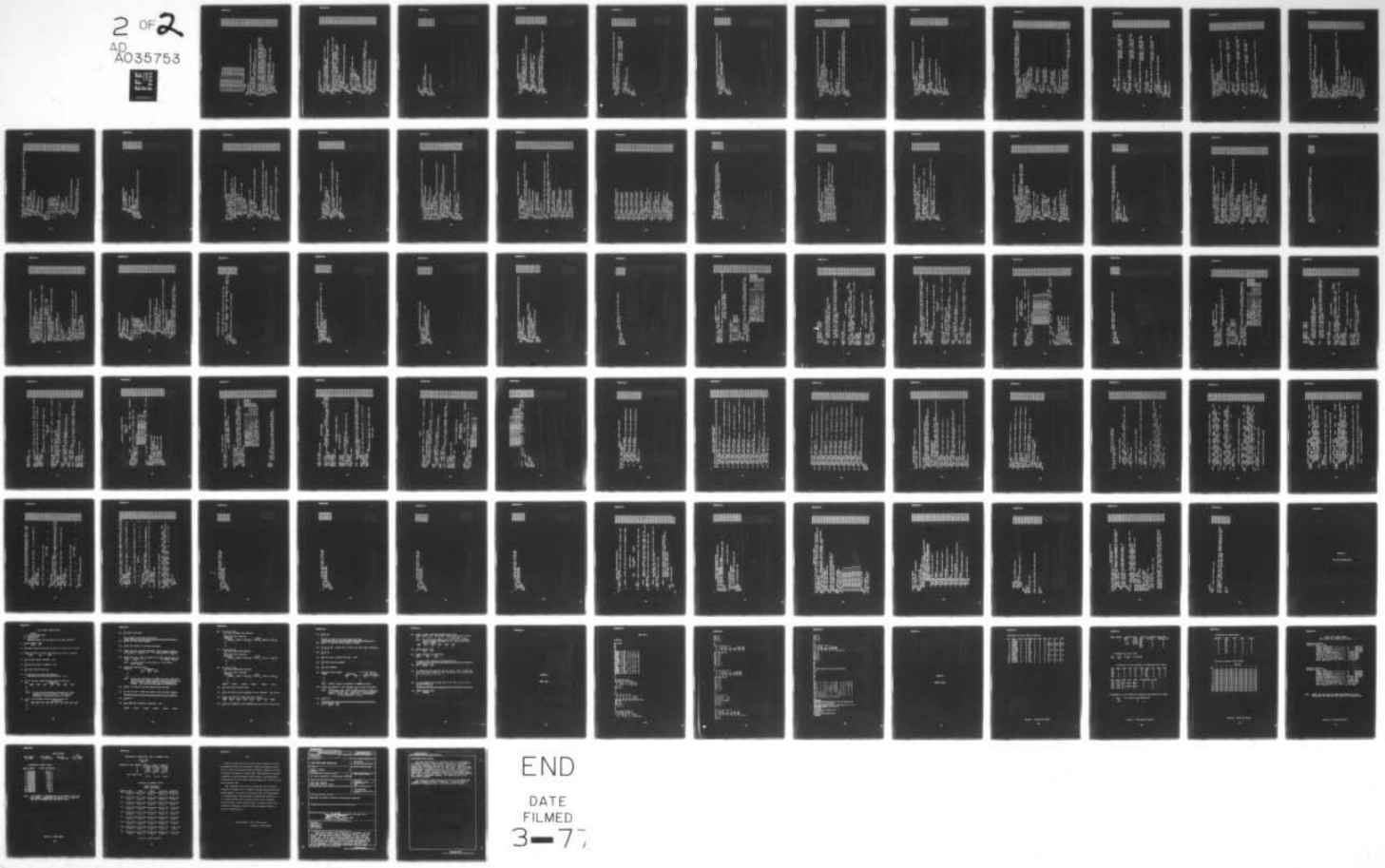
```
**                                                                    00005960
  EVENT CONFIRMED.MISSION GIVEN GTZ SAVING THE EVENT NOTICE           00005970
  DEFINE T AS A REAL VARIABLE                                         00005980
  CALL DISBAND.PARTIAL.TEAMS                                          00005990
  IF EVENT IS EXTERNAL                                                00006000
    READ GTZ                                                          00006010
    GO TO EXTERNAL                                                    00006020
  ELSE                                                                00006030
  IF WEEKDAY.F(TIME.V) LE VWHDD                                       00006040
    GO TO WEEKDAY                                                     00006050
  ELSE                                                                00006060
  SCHEDULE THIS CONFIRMED.MISSION(GTZ) AT TIME.V + 1                  00006070
**NOTE   INTERNAL MISSIONS MAY BE SCHEDULED FOR NON DUTY HOURS BUT     00006080
**       ONLY EXTERNAL MISSIONS CAN BE SCHEDULED ON NON DUTY DAYS.     00006090
  RETURN                                                              00006100
    *WEEKDAY*                                                         00006110
    *EXTERNAL*                                                        00006120
  ADD 1 TO TCNF(GTZ)                                                  00006130
  CREATE A MISSION CALLED GMZ                                         00006140
  LET MTPN(GMZ) = GTZ                                                 00006150
  LET MCEL(GMZ) = 1                                                   00006160
  LET MTOA(GMZ) = TIME.V                                              00006170
  LET MXCI(GMZ) = 0                                                   00006180
  IF EVENT IS INTERNAL                                                00006190
    FOR EACH AZK OF STRUCTURE(GTZ) DO                                 00006200
      CREATE AN ACTIVITY CALLED GAZ                                   00006210
      LET ATPC(GAZ) = ATPC(AZK)                                       00006220
      LET ANOA(GAZ) = ANOA(AZK)                                       00006230
      LET APOA(GAZ) = APOA(AZK)                                       00006240
      LET ATFC(GAZ) = ATFC(AZK)                                       00006250
      LET AICT(GAZ) = AICT(AZK)                                       00006260
      LET ANWR(GAZ) = ANWR(AZK)                                       00006270
      LET APAR(GAZ) = APAR(AZK)                                       00006280
      LET AFWL(GAZ) = AFWL(AZK)                                       00006290
      LET ASWL(GAZ) = ASWL(AZK)                                       00006300
```

```
      LET ATWL(GAZ) = ATWL(AZK)                                          00006310
      LET ACEL(GAZ) = ACEL(AZK)                                          00006320
      LET APMA(GAZ) = APMA(AZK)                                          00006330
      LET ALRL(GAZ) = ALRL(AZK)                                          00006340
      LET AURL(GAZ) = AURL(AZK)                                          00006350
      FILE GAZ IN ROUTING(GMZ)                                           00006360
    LOOP                                                                 00006370
    GO TO CNX.PROB                                                       00006380
  ELSE                                                                   00006390
**PERFORM A SPECIAL MISSION                                              00006400
  UNTIL MODE IS ALPHA DO                                                 00006410
    READ N                                                               00006420
    FOR EACH AZK IN STRUCTURE(GTZ) WITH ATPC(AZK) = N                    00006430
      FIND THE FIRST CASE                                                00006440
      IF NONE                                                            00006450
        PRINT 3 LINES WITH N AND GTZ THUS                                00006460
                                                                         00006470
  WARNING    ACTIVITY CODE *** IS NOT PART OF THE SEQUENCE OF            00006480
             TEST PHASE ***,     VERIFY THE INPUT                        00006490
      CREATE AN ACTIVITY CALLED GAZ                                      00006500
      READ ANOA(GAZ), APOA(GAZ), ANWR(GAZ),                              00006510
        AFWL(GAZ), ASWL(GAZ), ATWL(GAZ)                                  00006520
      LET ATPC(GAZ) = N                                                  00006530
      LET ATFC(GAZ) = KTFC(ANOA(GAZ))                                    00006540
      LET AICT(GAZ) = 0                                                  00006550
      LET APAR(GAZ) = KPAR(ANOA(GAZ))                                    00006560
      LET ACEL(GAZ) = 1                                                  00006570
      LET APMA(GAZ) = KPMA(ANOA(GAZ))                                    00006580
      LET ALRL(GAZ) = KLRL(ANOA(GAZ))                                    00006590
      LET AURL(GAZ) = KURL(ANOA(GAZ))                                    00006600
      FILE GAZ IN ROUTING(GMZ)                                           00006610
      CALL MY.LIST(5,GMZ)**ACTIVITIES IN ROUTING(GMZ)                    00006620
      GO TO LOOP                                                         00006630
    ELSE                                                                 00006640
    CREATE AN ACTIVITY CALLED GAZ                                        00006650
```

AD-A035 753 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 5/1
JOINT TEST FORCE SIMULATION.(U)
DEC 76 J P EGGERS

UNCLASSIFIED GOR/SM/76D-7 NL

2 OF 2
AD
A035753



END

DATE
FILMED
3-77

```
    LET ATPC(GAZ) = N                                                    00006660
    LET ANOA(GAZ) = ANOA(AZK)                                            00006670
    LET APOA(GAZ) = APOA(AZK)                                            00006680
    LET ATFC(GAZ) = ATFC(AZK)                                            00006690
    LET AICT(GAZ) = AICT(AZK)                                            00006700
    LET ANWR(GAZ) = ANWR(AZK)                                            00006710
    LET APAR(GAZ) = APAR(AZK)                                            00006720
    LET AFWL(GAZ) = AFWL(AZK)                                            00006730
    LET ASWL(GAZ) = ASWL(AZK)                                            00006740
    LET ATWL(GAZ) = ATWL(AZK)                                            00006750
    LET ACEL(GAZ) = ACEL(AZK)                                            00006760
    LET APMA(GAZ) = APMA(AZK)                                            00006770
    LET ALRL(GAZ) = ALRL(AZK)                                            00006780
    LET AURL(GAZ) = AURL(AZK)                                            00006790
    FILE GAZ IN ROUTING(GMZ)                                             00006800
      'LOOP'                                                             00006810
LOOP                                                                     00006820
  'CNX.PROB'                                                             00006830
FOR EACH FACTOR OF TCANCELLATION.PROBABILITY(GTZ) DO                     00006840
  CREATE AN XFACTOR                                                      00006850
  LET XUPL(XFACTOR) = XUPL(FACTOR)                                       00006860
  FILE XFACTOR IN MCANCELLATION.PROBABILITY(GMZ)                         00006870
LOOP                                                                     00006880
LET MNAP(GMZ) = APOA(F.ROUTING(GMZ))**RANKING CODE IN QUEUE              00006890
LET MNAC(GMZ) = ATPC(F.ROUTING(GMZ))**RANKING CODE IN QUEUE              00006900
CALL VALIDATE.NUMBER.OF.WORKERS.REQUESTED GIVING GMZ                     00006910
CALL DETERMINE.PREMISSION.CANCELLATION GIVING GMZ                        00006920
CALL CALCULATE.MISSION.EFFECTIVENESS GIVING GMZ                          00006930
IF MXCI(GMZ) = 1                                                         00006940
  DESTROY THIS MISSION CALLED GMZ                                        00006950
  CALL BUILD.PARTIAL.TEAMS**REBUILD PARTIAL TEAMS                        00006960
  IF EVENT IS EXTERNAL                                                   00006970
    DESTROY THIS CONFIRMED.MISSION                                       00006980
    RETURN                                                               00006990
  ELSE                                                                   00007000
```

```
    CALL MSN.SCHEDULE(GTZ)                                          00007010
    RETURN                                                          00007020
  ELSE                                                              00007030
  IF MNAP(GMZ) = 1 AND MTOA(GMZ) + VRMPT GT TIME.V                  00007040
    FILE GMZ IN WAIT.FOR.MISSION.TIME                               00007050
    SCHEDULE AN ACCOMPLISH.PR1.MISSION AT MTOA(GMZ)+VRMPT           00007060
    GO COUNT                                                        00007070
  ELSE                                                              00007080
**PERFORM PRIORITY MISSIONS SCHEDULED DURING NON DUTY HOURS         00007090
  IF WEEKDAY.F(TIME.V) GT VWWDD OR **WEEKEND                        00007100
    FRAC.F(TIME.V) GE VEDDT/24 OR **AFTER DUTY HOURS                00007110
    FRAC.F(TIME.V) LT VBDDT/24 **BEFORE DUTY HOURS                  00007120
    FILE GMZ IN QUEUE                                               00007130
    IF MNAP(GMZ) = 1 OR                                             00007140
       MNAP(GMZ) = 2                                                00007150
      CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                00007160
    ALWAYS                                                          00007170
    GO COUNT                                                        00007180
  ELSE                                                              00007190
  LET T = FRAC.F(TIME.V) * 24                                       00007200
  IF VBB1T LT T LE VEB1T OR **BREAK TIME                            00007210
     VBLBT LT T LE VELBT OR                                         00007220
     VBB2T LT T LE VEB2T                                            00007230
    FILE GMZ IN QUEUE                                               00007240
    CALL CHECK.QUEUE                                                00007250
    GO COUNT                                                        00007260
  ELSE                                                              00007270
  CALL ATTEND.TO.MISSION GIVING GMZ **NORMAL DUTY                   00007280
  CALL CHECK.QUEUE**REASSIGN WORKERS IF PREEMPT OCCURRED            00007290
    *COUNT*                                                         00007300
  IF TCMP(GTZ)+LACM(GTZ) GE TREQ(GTZ)                               00007310
**CURRENT MISSION IS COUNTED IN LACM                                00007320
    ADD 1 TO GTZ                                                    00007330
    IF GTZ GT N.TEST.PHASE                                          00007340
      DESTROY THIS CONFIRMED.MISSION                                00007350
```

```
    RETURN                                                          00007360
  ELSE                                                              00007370
ALWAYS                                                              00007380
IF EVENT IS EXTERNAL                                                00007390
  DESTROY THIS CONFIRMED.MISSION                                    00007400
  RETURN                                                            00007410
ELSE                                                                00007420
CALL MSN.SCHEDULE(GTZ)                                              00007430
RETURN                                                              00007440
END                                                                 00007450
```

```
**                                                                    00007460
                    ROUTINE MSN.SCHEDULE(GTZ)                         00007470
  CALL MY.TRACE(19,GTZ)                                               00007480
  IF TDBM(GTZ) LT 1.0                                                 00007490
    IF FRAC.F(TIME.V+TDBM(GTZ)) GE VBDDT/24 AND                       00007500
       FRAC.F(TIME.V+TDBM(GTZ)) LT VEDDT/24                           00007510
      SCHEDULE THE CONFIRMED.MISSION(GTZ) AT TIME.V+TDBM(GTZ)         00007520
      RETURN                                                          00007530
    ELSE                                                              00007540
    SCHEDULE THE CONFIRMED.MISSION(GTZ) AT                            00007550
      TRUNC.F(TIME.V+1) + VBDDT/24                                    00007560
**    TOMMORROW + BEGINNING OF DUTY HOURS                             00007570
    RETURN                                                            00007580
  ELSE                                                                00007590
  SCHEDULE THE CONFIRMED.MISSION(GTZ) AT                              00007600
    TRUNC.F(TIME.V) + TDBM(GTZ) + TTMC(GTZ)/24                        00007610
** BEGIN OF CURRENT DAY + DAYS BEFORE NEXT MISSION + TIME OF DAY      00007620
  RETURN                                                              00007630
  END                                                                 00007640
```

```
..                                                                          00007650
        ROUTINE TO VALIDATE.NUMBER.OF.WORKERS.REQUESTED GIVEN RMZ           00007660
CALL MY.TRACE(8,RMZ)                                                        00007670
FOR EVERY RAZ IN ROUTING(RMZ) DO                                            00007680
  FOR EVERY WORKER CALLED RWZ WITH WLEV(RWZ) = AFWL(RAZ) OR                 00007690
                                   WLEV(RWZ) = ASWL(RAZ) OR                 00007700
                                   WLEV(RWZ) = ATWL(RAZ)                    00007710
    ADD 1 TO I                                                              00007720
  IF ANWR(RAZ) GT I                                                         00007730
    CALL ERROR.PACKAGE(4,RMZ)                                               00007740
  ALWAYS                                                                    00007750
  LET I = 0                                                                 00007760
LOOP                                                                        00007770
RETURN                                                                      00007780
END                                                                         00007790
```

```
..                                                                    00007800
        ROUTINE TO DETERMINE.PREMISSION.CANCELLATION GIVEN FMZ        00007810
CALL MY.TRACE(3,FMZ)                                                  00007820
FOR EVERY XFACTOR IN MCANCELLATION.PROBABILITY(FMZ) DO                00007830
  IF RANDOM.F(7) LE XUPL(XFACTOR)                                     00007840
    LET MXCI(FMZ) = 1                                                 00007850
  ALWAYS                                                              00007860
LOOP                                                                  00007870
RETURN                                                                00007880
END                                                                   00007890
```

```
..                                                                          00007900
        ROUTINE TO CALCULATE.MISSION.EFFECTIVENESS GIVEN HMZ                00007910
CALL MY.TRACE(4,HMZ)                                                        00007920
FOR EACH HAZ IN ROUTING(HMZ) DO                                             00007930
  IF RANDOM.F(7) LE APMA(HAZ)/100                                           00007940
    LET ACEL(HAZ)=MCEL(HMZ)-UNIFORM.F(ALRL(HAZ)/100,AURL(HAZ)/100,8)        00007950
    LET MCEL(HMZ) = ACEL(HAZ)                                               00007960
    JUMP AHEAD                                                              00007970
  ELSE                                                                      00007980
  LET ACEL(HAZ) = MCEL(HMZ)                                                 00007990
    HERE                                                                    00008000
LOOP                                                                        00008010
IF MCEL(HMZ) LE 0                                                           00008020
  LET MXCI(HMZ) = 1                                                         00008030
  PRINT 3 LINES WITH MTPN(HMZ) AND TIME.V THUS                              00008040
                                                                            00008050
  A MISSION IN TEST PHASE *** WAS ELIMINATED AT ****.**** DAYS              00008060
  BECAUSE THE TOTAL MISSION EFFECTIVENESS REDUCTIONS EXCEEDED 100%          00008070
ALWAYS                                                                      00008080
RETURN                                                                      00008090
END                                                                         00008100
```

```
**                                                                    00008110
                    ROUTINE TO ATTEND.TO.MISSION GIVEN IMZ          00008120
CALL MY.TRACE(5,IMZ)                                                00008130
LET IAZ = F.ROUTING(IMZ)                                            00008140
IF MNAP(IMZ) = 1                                                    00008150
  IF MTOA(IMZ) + VRMPT GT TIME.V                                    00008160
    FILE IMZ IN WAIT.FOR.MISSION.TIME                               00008170
    SCHEDULE AN ACCOMPLISH.PR1.MISSION AT MTOA(IMZ) + VRMPT         00008180
    RETURN                                                          00008190
  ELSE                                                              00008200
**BREAK TIME                                                        00008210
  IF ANWR(IAZ) LE SUM.WORKERS(2,IMZ,0)                              00008220
    JUMP AHEAD                                                      00008230
  ELSE                                                              00008240
ALWAYS                                                              00008250
IF ANWR(IAZ) LE SUM.WORKERS(1,IMZ,0)                                00008260
  HERE                                                              00008270
  CALL ALLOCATION GIVING IMZ                                        00008280
  RETURN                                                            00008290
ELSE                                                                00008300
FILE IMZ IN QUEUE                                                   00008310
RETURN                                                              00008320
END                                                                 00008330
```

```
**                                                                    00008340
                          ROUTINE SUM.WORKERS(NUMB,JMZ,PART.TEAM)     00008350
**CONSIDER ALL COMBINATIONS OF WORKER LEVEL PREFERENCES IN DETERMINING 00008360
**THE NUMBER OF WORKERS CURRENTLY AVAILABLE TO PERFORM AN ACTIVITY.    00008370
  CALL MY.TRACE(6,JMZ)                                                00008380
  LET STATUS = 1                                                      00008390
  LET JAZ = F.ROUTING(JMZ)                                            00008400
 GO TO 1,2,3 PER NUMB                                                 00008410
  *1* **WORKERS WHO ARE IDLE                                          00008420
      IF AFWL(JAZ) = ASWL(JAZ)                                        00008430
        IF ASWL(JAZ) = ATWL(JAZ)                                      00008440
          LET X = LNIW(AFWL(JAZ))                                     00008450
          GO PR1                                                      00008460
        ELSE                                                          00008470
        LET X = LNIW(AFWL(JAZ)) +                                     00008480
                LNIW(ATWL(JAZ))                                       00008490
        GO PR1                                                        00008500
      ELSE                                                            00008510
      IF ASWL(JAZ) = ATWL(JAZ)                                        00008520
        LET X = LNIW(AFWL(JAZ)) +                                     00008530
                LNIW(ASWL(JAZ))                                       00008540
        GO PR1                                                        00008550
      ELSE                                                            00008560
      LET X = LNIW(ASWL(JAZ)) +                                       00008570
              LNIW(ATWL(JAZ))                                         00008580
      IF AFWL(JAZ) = ATWL(JAZ)                                        00008590
        GO PR1                                                        00008600
      ELSE                                                            00008610
      ADD LNIW(AFWL(JAZ)) TO X                                        00008620
      GO PR1                                                          00008630
  *2* **WORKERS WHO ARE NOT BUSY                                      00008640
      IF AFWL(JAZ) = ASWL(JAZ)                                        00008650
        IF ASWL(JAZ) = ATWL(JAZ)                                      00008660
          FOR EVERY WORKER CALLED JWZ WITH WLEV(JWZ) = AFWL(JAZ) AND  00008670
                                           WSTA(JWZ) NE STATUS        00008680
```

```
            ADD 1 TO X                                                          00008690
            GO TO PR1                                                           00008700
          ELSE                                                                  00008710
          FOR EVERY WORKER CALLED JWZ WITH WLEV(JWZ) = AFWL(JAZ) OR             00008720
                                           WLEV(JWZ) = ATWL(JAZ) AND            00008730
                                           WSTA(JWZ) NE STATUS                  00008740
            ADD 1 TO X                                                          00008750
          GO TO PR1                                                             00008760
        ELSE                                                                    00008770
        IF ASWL(JAZ) = ATWL(JAZ)                                                00008780
          FOR EVERY WORKER CALLED JWZ WITH WLEV(JWZ) = AFWL(JAZ) OR             00008790
                                           WLEV(JWZ) = ASWL(JAZ) AND            00008800
                                           WSTA(JWZ) NE STATUS                  00008810
            ADD 1 TO X                                                          00008820
          GO TO PR1                                                             00008830
        ELSE                                                                    00008840
        FOR EVERY WORKER CALLED JWZ WITH WLEV(JWZ) = ASWL(JAZ) OR               00008850
                                         WLEV(JWZ) = ATWL(JAZ) AND              00008860
                                         WSTA(JWZ) NE STATUS                    00008870
          ADD 1 TO X                                                            00008880
        IF AFWL(JAZ) = ATWL(JAZ)                                                00008890
          GO TO PR1                                                             00008900
        ELSE                                                                    00008910
        FOR EVERY WORKER CALLED JWZ WITH WLEV(JWZ) = AFWL(JAZ) AND              00008920
                                         WSTA(JWZ) NE STATUS                    00008930
          ADD 1 TO X                                                            00008940
        GO TO PR1                                                               00008950
*3*   **WORKERS WHO ARE NOT WORKING OVERTIME                                    00008960
        LET STATUS = 4                                                          00008970
        GO TO 2                                                                 00008980
*PR1*                                                                           00008990
IF PART.TEAM = 1                                                                00009000
  RETURN WITH X                                                                 00009010
ELSE                                                                            00009020
IF APOA(JAZ) = 1                                                                00009030
```

```
  FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH                                    00009040
    APOA(F.ROUTING(EAMN(K))) NE 1 DO                                         00009050
    LET NMZ = EAMN(K)                                                        00009060
    LET NAZ = F.ROUTING(NMZ)                                                 00009070
    IF AFWL(JAZ) = ASWL(JAZ)                                                 00009080
      IF ASWL(JAZ) = ATWL(JAZ)                                               00009090
        FOR EVERY NWZ IN TEAM(NAZ) WITH WLEV(NWZ) = AFWL(JAZ)                00009100
          ADD 1 TO X                                                         00009110
        JUMP AHEAD                                                           00009120
      ELSE                                                                   00009130
      FOR EVERY NWZ IN TEAM(NAZ) WITH WLEV(NWZ) = AFWL(JAZ) OR               00009140
                                      WLEV(NWZ) = ATWL(JAZ)                  00009150
        ADD 1 TO X                                                           00009160
      JUMP AHEAD                                                             00009170
    ELSE                                                                     00009180
    IF ASWL(JAZ) = ATWL(JAZ)                                                 00009190
        FOR EVERY NWZ IN TEAM(NAZ) WITH WLEV(NWZ) = AFWL(JAZ) OR             00009200
                                        WLEV(NWZ) = ASWL(JAZ)                00009210
        ADD 1 TO X                                                           00009220
      JUMP AHEAD                                                             00009230
    ELSE                                                                     00009240
    FOR EVERY NWZ IN TEAM(NAZ) WITH WLEV(NWZ) = ASWL(JAZ) OR                 00009250
                                    WLEV(NWZ) = ATWL(JAZ)                    00009260
      ADD 1 TO X                                                             00009270
    IF AFWL(JAZ) = ATWL(JAZ)                                                 00009280
      JUMP AHEAD                                                             00009290
    ELSE                                                                     00009300
      FOR EVERY NWZ IN TEAM(NAZ) WITH WLEV(NWZ) = AFWL(JAZ)                  00009310
      ADD 1 TO X                                                             00009320
  LOOP                                                                       00009330
    HERE                                                                     00009340
ALWAYS                                                                       00009350
RETURN WITH X                                                                00009360
END                                                                          00009370
```

```
**                                                                    00009380
                    ROUTINE FOR ALLOCATION GIVEN PMZ AND PART.TEAM   00009390
  CALL MY.TRACE(7,PMZ)                                               00009400
  DEFINE VAL AS A REAL VARIABLE                                      00009410
  IF ANWR(F.ROUTING(PMZ)) = 0 **MANPOWER REQ IS EXTERNAL             00009420
    GO TO WORKER.FOUND                                               00009430
  ELSE                                                               00009440
  LET PAZ = F.ROUTING(PMZ)                                           00009450
  *PREEMPT.COMPLETED.LOOK.AGAIN*                                     00009460
  LET J=0                                                            00009470
  LET I = AFWL(PAZ)                                                  00009480
  *LOOK.AGAIN*                                                       00009490
  IF I=0                                                             00009500
    GO TO U                                                          00009510
  ELSE                                                               00009520
  *MORE.REQUIRED*                                                    00009530
  LET PWZ = 0                                                        00009540
  FOR EVERY WORKER CALLED WZI WITH WLEV(WZI) = I AND WSTA(WZI) = IDLE 00009550
    COMPUTE PWZ = MIN(WZI) OF TOBSERVED.STATE(WZI,2)                 00009560
  IF PWZ NE 0                                                        00009570
    FILE PWZ IN TEAM(PAZ)                                            00009580
    LET WSTA(PWZ) = BUSY                                             00009590
    SUBTRACT 1 FROM LNIW(WLEV(PWZ))                                  00009600
    SUBTRACT 1 FROM ANWR(PAZ)                                        00009610
    IF ANWR(PAZ) LT 0                                                00009620
      CALL ERROR.PACKAGE(2,PMZ)                                      00009630
    ALWAYS                                                           00009640
    IF ANWR(PAZ) = 0                                                 00009650
      GO TO WORKER.FOUND                                             00009660
    ELSE                                                             00009670
    GO TO MORE.REQUIRED                                              00009680
  ALWAYS                                                             00009690
**BREAK TIME ACTIVITIES                                              00009700
  IF APOA(PMZ) = 1                                                   00009710
    LET PWZ = 0                                                      00009720
```

```
    FOR EVERY WORKER CALLED WZI WITH WLEV(WZI) = I AND WSTA(WZI) NE BUSY00009730
      COMPUTE PWZ = MIN(WZI) OF TOBSERVED.STATE(WZI,2)                00009740
    IF PWZ NE 0                                                       00009750
      FILE PWZ IN TEAM(PAZ)                                           00009760
      LET WSTA(PWZ) = BUSY                                            00009770
      SUBTRACT 1 FROM ANWR(PAZ)                                       00009780
      IF ANWR(PAZ) LT 0                                               00009790
        CALL ERROR.PACKAGE(2,PMZ)                                     00009800
      ALWAYS                                                          00009810
      IF ANWR(PAZ) = 0                                                00009820
        GO TO WORKER.FOUND                                            00009830
      ELSE                                                            00009840
      GO TO MORE.REQUIRED                                             00009850
    ELSE                                                              00009860
  ALWAYS                                                              00009870
  ADD 1 TO J                                                          00009880
  GO TO W, X, Y, V PER J                                              00009890
    *W* LET I = ASWL(PAZ)                                             00009900
          GO TO LOOK.AGAIN                                            00009910
    *X* LET I = ATWL(PAZ)                                             00009920
          GO TO LOOK.AGAIN                                            00009930
    *Y* IF PART.TEAM = 1                                              00009940
          GO TO WORKER.FOUND                                          00009950
        ELSE                                                          00009960
        IF MNAP(PMZ) = 1                                              00009970
          CALL PREEMPT                                                00009980
          ADD 1 TO L                                                  00009990
          IF L GT N.WORKER                                            00010000
            GO TO V                                                   00010010
          ELSE                                                        00010020
          GO TO PREEMPT.COMPLETED.LOOK.AGAIN                          00010030
        ELSE                                                          00010040
        GO TO V                                                       00010050
    *V* UNTIL TEAM(PAZ) IS EMPTY DO                                   00010060
          REMOVE THE FIRST PWZ FROM TEAM(PAZ)                         00010070
```

```
            ADD 1 TO LNIW(WLEV(PWZ))                                    00010080
            ADD 1 TO ANWR(PAZ)                                          00010090
            LET WSTA(PWZ) = IDLE                                        00010100
          LOOP                                                          00010110
          FILE PWZ IN QUEUE                                             00010120
          RETURN                                                        00010130
      *U*CALL ERROR.PACKAGE(3,PWZ)                                      00010140
    *WORKER.FOUND*                                                      00010150
    LET VAL = VALUE(PWZ)                                                00010160
    SCHEDULE AN EO.ACTIVITY GIVING PWZ IN VAL MINUTES                   00010170
    RETURN                                                              00010180
    END                                                                 00010190
```

```
**                                                                    00010200
                          EVENT ACCOMPLISH.PR1.MISSION                00010210
DEFINE T AS A REAL VARIABLE                                           00010220
LET T = FRAC.F(TIME.V) * 24                                           00010230
CALL DISBAND.PARTIAL.TEAMS                                            00010240
FOR EVERY SMZ IN WAIT.FOR.MISSION.TIME WITH                           00010250
  MTOA(SMZ) + VRMPT LE TIME.V DO                                      00010260
  REMOVE THIS SMZ FROM WAIT.FOR.MISSION.TIME                          00010270
  IF VBB1T LT T LE VEB1T OR                                           00010280
     VBLBT LT T LE VELBT OR                                           00010290
     VBB2T LT T LE VEB2T                                              00010300
    THEN IF WEEKDAY.F(TIME.V) LE VWWDD                                00010310
      FILE SMZ IN QUEUE                                               00010320
      CALL CHECK.QUEUE                                                00010330
      JUMP AHEAD                                                      00010340
  ELSE                                                                00010350
  IF WEEKDAY.F(TIME.V) GT VWWDD                                       00010360
    FILE SMZ IN QUEUE                                                 00010370
    IF VWEAM = "YES"                                                  00010380
      CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                  00010390
      JUMP AHEAD                                                      00010400
    ELSE                                                              00010410
    RETURN                                                            00010420
  ELSE                                                                00010430
  IF FRAC.F(TIME.V) GE VEDDT/24 OR FRAC.F(TIME.V) LT VBDDT/24         00010440
    FILE SMZ IN QUEUE                                                 00010450
    CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                    00010460
    JUMP AHEAD                                                        00010470
  ELSE                                                                00010480
  CALL ATTEND.TO.MISSION GIVING SMZ                                   00010490
     HERE                                                             00010500
LOOP                                                                  00010510
CALL CHECK.QUEUE**REASSIGN WORKERS IF PREEMPT OCCURRED                00010520
RETURN                                                                00010530
END                                                                   00010540
```

```
**                                                                    00010550
                          ROUTINE TO PREEMPT                          00010560
  CALL MY.TRACE(9)                                                    00010570
  LET REASON = IDLE                                                   00010580
  IF FRAC.F(TIME.V) GE VEDDT/24 OR                                    00010590
     FRAC.F(TIME.V) LT VBDDT/24                                       00010600
    LET REASON = OFF.DUTY                                             00010610
  ALWAYS                                                              00010620
  FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MNAP(EAMN(K)) NE 1          00010630
    FIND THE FIRST CASE                                               00010640
    IF FOUND                                                          00010650
      CALL INTERRUPT.AND.FILE GIVING K AND REASON                     00010660
    ALWAYS                                                            00010670
**PREEMPT ONLY 1 MISSION                                              00010680
  RETURN                                                              00010690
  END                                                                 00010700
```

```
..                                                                          00010710
            ROUTINE TO INTERRUPT.AND.FILE GIVEN K AND REASON                00010720
CALL MY.TRACE(10,EAMN(K))                                                   00010730
DEFINE AV.FACTOR AS A REAL VARIABLE                                         00010740
LET AV.FACTOR = 1                                                           00010750
LET PART.TEAM.FACTOR = 1                                                    00010760
REMOVE K FROM EV.S(I.EO.ACTIVITY)                                           00010770
LET UMZ = EAMN(K)                                                           00010780
LET UAZ = F.ROUTING(UMZ)                                                    00010790
IF TEAM(UAZ) IS NOT EMPTY                                                   00010800
LET PART.TEAM.FACTOR = (ANWR(UAZ)+N.TEAM(UAZ))/N.TEAM(UAZ)                  00010810
ALWAYS                                                                      00010820
UNTIL TEAM(UAZ) IS EMPTY DO                                                 00010830
  REMOVE THE FIRST UWZ FROM TEAM(UAZ)                                       00010840
  COMPUTE AV.FACTOR = AVG OF SKILL.FACTOR(UWZ,ANOA(UAZ))                    00010850
  ADD 1 TO ANWR(UAZ)                                                        00010860
  LET WSTA(UWZ) = REASON                                                    00010870
  IF WSTA(UWZ) = IDLE                                                       00010880
    ADD 1 TO LNIW(WLEV(UWZ))                                                00010890
  ALWAYS                                                                    00010900
LOOP                                                                        00010910
LET AICT(UAZ) = (TIME.A(K) - TIME.V)/(AV.FACTOR * PART.TEAM.FACTOR)         00010920
FILE UMZ IN QUEUE                                                           00010930
DESTROY EO.ACTIVITY CALLED K                                                00010940
RETURN                                                                      00010950
END                                                                         00010960
```

```
..                                                                    00010970
                        ROUTINE VALUE GIVEN YMZ                       00010980
  CALL MY.TRACE(11,YMZ)                                               00010990
  LET YAZ = F.ROUTING(YMZ)                                            00011000
  LET YTZ = MTPN(YMZ)                                                 00011010
  DEFINE A, B, SAMPLE, AV.FACTOR, AND VAL AS REAL VARIABLES           00011020
  DEFINE C, D, AND E AS INTEGER VARIABLES                             00011030
  IF AICT(YAZ) GT 0                                                   00011040
    LET SAMPLE = AICT(YAZ)*24*60                                      00011050
    LET AICT(YAZ) = 0                                                 00011060
    GO TO PROFICIENCY.OF.WORKER                                       00011070
  ELSE                                                                00011080
  LET A = 1ARG(YTZ,ANOA(YAZ))                                         00011090
  LET B = 2ARG(YTZ,ANOA(YAZ))                                         00011100
  LET C = 10                                                          00011110
  LET E = ATFC(YAZ)                                                   00011120
  IF E LT 0                                                           00011130
    GO TO INTEGER.STEP, REAL.STEP OR CONTINUOUS PER ABS.F(E)          00011140
  ELSE                                                                00011150
  GO TO BETA, BINOMIAL, CONSTANT, ERLANG, EXPONENTIAL, GAMMA, NORMAL, 00011160
      POISSON, UNIFORM OR WEIBULL PER E                               00011170
  *BETA*                                                              00011180
    LET SAMPLE = BETA.F(A,B,C)                                        00011190
    GO TO PROFICIENCY.OF.WORKER                                       00011200
  *BINOMIAL*                                                          00011210
    LET D = TRUNC.F(A)                                                00011220
    LET SAMPLE = BINOMIAL.F(D,B,C)                                    00011230
    GO TO PROFICIENCY.OF.WORKER                                       00011240
  *CONSTANT*                                                          00011250
    LET SAMPLE = A                                                    00011260
    GO TO PROFICIENCY.OF.WORKER                                       00011270
  *ERLANG*                                                            00011280
    LET SAMPLE = ERLANG.F(A,B,C)                                      00011290
    GO TO PROFICIENCY.OF.WORKER                                       00011300
  *EXPONENTIAL*                                                       00011310
```

```
  LET SAMPLE = EXPONENTIAL.F(A,C)                                    00011320
  GO TO PROFICIENCY.OF.WORKER                                        00011330
*GAMMA*                                                              00011340
  LET SAMPLE = GAMMA.F(A,B,C)                                        00011350
  GO TO PROFICIENCY.OF.WORKER                                        00011360
*NORMAL*                                                             00011370
  LET SAMPLE = NORMAL.F(A,B,C)                                       00011380
  GO TO PROFICIENCY.OF.WORKER                                        00011390
*POISSON*                                                            00011400
  LET SAMPLE = POISSON.F(A,C)                                        00011410
  GO TO PROFICIENCY.OF.WORKER                                        00011420
*UNIFORM*                                                            00011430
  LET SAMPLE = UNIFORM.F(A,B,C)                                      00011440
  GO TO PROFICIENCY.OF.WORKER                                        00011450
*WEIBULL*                                                            00011460
  LET SAMPLE = WEIBULL.F(A,B,C)                                      00011470
  GO TO PROFICIENCY.OF.WORKER                                        00011480
*INTEGER.STEP*                                                       00011490
  LET D = TRUNC.F(B)                                                 00011500
  **FOR INTEGER.STEP.FUNCTION = D                                    00011510
    LET SAMPLE = ONE(D)                                              00011520
  GO TO PROFICIENCY.OF.WORKER                                        00011530
*REAL.STEP*                                                          00011540
  LET D = TRUNC.F(B)                                                 00011550
  **FOR REEL.STEP.FUNCTION = D                                       00011560
    LET SAMPLE = TWO(D)                                              00011570
  GO TO PROFICIENCY.OF.WORKER                                        00011580
*CONTINUOUS*                                                         00011590
  LET D = TRUNC.F(B)                                                 00011600
  **FOR CONTINUOUS.FUNCTION = D                                      00011610
    LET SAMPLE = THREE(D)                                            00011620
  GO TO PROFICIENCY.OF.WORKER                                        00011630
*PROFICIENCY.OF.WORKER*                                              00011640
IF TEAM(YAZ) IS EMPTY                                                00011650
  RETURN WITH SAMPLE                                                 00011660
```

```
ELSE                                                                    00011670
FOR EVERY YWZ IN TEAM(YAZ)                                              00011680
  COMPUTE AV.FACTOR = AVG OF SKILL.FACTOR(YWZ,ANOA(YAZ))                00011690
LET PART.TEAM.FACTOR = (ANWR(YAZ)+N.TEAM(YAZ))/N.TEAM(YAZ)              00011700
LET VAL = SAMPLE * AV.FACTOR * PART.TEAM.FACTOR                         00011710
RETURN WITH VAL                                                         00011720
END                                                                     00011730
```

```
..
          ROUTINE LOOK.AHEAD.COMPLETE.MISSIONS GIVEN QTZ                    00011740
CALL MY.TRACE(12,QTZ)                                                       00011750
DEFINE SUMF AS A REAL VARIABLE                                              00011760
LET SUMF = 0                                                                00011770
FOR EVERY QMZ IN QUEUE WITH MTPN(QMZ) = QTZ                                 00011780
  ADD ACEL(L.ROUTING(QMZ)) TO SUMF                                          00011790
FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MTPN(EAMN(K)) = QTZ                 00011800
  ADD ACEL(L.ROUTING(EAMN(K))) TO SUMF                                      00011810
FOR EVERY XMZ IN WAIT.FOR.MISSION.TIME WITH MTPN(XMZ)=QTZ                   00011820
  ADD ACEL(L.ROUTING(XMZ)) TO SUMF                                          00011830
RETURN WITH SUMF                                                            00011840
                                                                            00011850
```

```
**                                                                    00011870
                    EVENT TAKE.BREAK SAVING THE EVENT NOTICE         00011880
SCHEDULE THIS TAKE.BREAK IN 24 HOURS                                 00011890
IF WEEKDAY.F(TIME.V) GT VWWDD                                        00011900
  RETURN **OFFICE IS CLOSED FOR THE WEEKEND                          00011910
ELSE                                                                 00011920
CALL DISBAND.PARTIAL.TEAMS                                           00011930
FOR EVERY WORKER CALLED WZL WITH WSTA(WZL) = IDLE DO                 00011940
  LET WSTA(WZL) = BREAK                                              00011950
  SUBTRACT 1 FROM LNIW(WLEV(WZL))                                    00011960
LOOP                                                                 00011970
FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MNAP(EAMN(K)) NE 1 DO        00011980
  LET REASON = BREAK                                                 00011990
  CALL INTERRUPT.AND.FILE GIVING K AND REASON                        00012000
LOOP                                                                 00012010
RETURN                                                               00012020
END                                                                  00012030
```

```
**                                                                    00012040
                        EVENT EO.ACTIVITY GIVEN MZB                   00012050
  DEFINE T AS A REAL VARIABLE                                         00012060
  CALL DISBAND.PARTIAL.TEAMS                                          00012070
  REMOVE THE FIRST AZB FROM ROUTING(MZB)                              00012080
  LET MNAP(MZB) = APOA(F.ROUTING(MZB))**RANKING CODE IN QUEUE         00012090
  LET MNAC(MZB) = ATPC(F.ROUTING(MZB))**RANKING CODE IN QUEUE         00012100
  UNTIL TEAM(AZB) IS EMPTY DO                                         00012110
    REMOVE THE FIRST WZB FROM TEAM(AZB)                               00012120
     IF FRAC.F(TIME.V) * 24 GE VEDDT OR                               00012130
        FRAC.F(TIME.V) * 24 LT VBDDT                                  00012140
      GO TO OFF.DUTY                                                  00012150
    ELSE                                                              00012160
    IF WEEKDAY.F(TIME.V) GT VWWDD                                     00012170
      *OFF.DUTY*                                                      00012180
      LET WSTA(WZB) = OFF.DUTY                                        00012190
      GO LOOP                                                         00012200
    ELSE                                                              00012210
    LET T = FRAC.F(TIME.V) * 24                                       00012220
    IF VBB1T LT T LE VEB1T OR                                         00012230
       VBLBT LT T LE VELBT OR                                         00012240
       VBB2T LT T LE VEB2T                                            00012250
      LET WSTA(WZB) = BREAK                                           00012260
      GO LOOP                                                         00012270
    ELSE                                                              00012280
    LET WSTA(WZB) = IDLE                                              00012290
    ADD 1 TO LNIW(WLEV(WZB))                                          00012300
      *LOOP*                                                          00012310
  LOOP                                                                00012320
  DESTROY THIS ACTIVITY CALLED AZB                                    00012330
  IF ROUTING(MZB) IS EMPTY                                            00012340
    LET TDCD(MTPN(MZB)) = TIME.V                                      00012350
    DESTROY THIS MISSION CALLED MZB                                   00012360
    GO CHECK.QUEUE                                                    00012370
  ELSE                                                                00012380
```

```
  FILE MZB IN QUEUE **INSURES THAT LOW CODED ACTIVITIES OF                00012390
**                          ALL MISSSIONS ARE PERFORMED FIRST.            00012400
  *CHECK.QUEUE*                                                           00012410
  IF WEEKDAY.F(TIME.V) GT VWHOD                                           00012420
    THEN IF VWEAM = "NO"                                                  00012430
    RETURN                                                                00012440
  ELSE                                                                    00012450
  CALL CHECK.QUEUE                                                        00012460
  RETURN                                                                  00012470
  END                                                                     00012480
```

```
**                                                                    00012490
                    EVENT EO.DUTY.HOURS SAVING THE EVENT NOTICE      00012500
  SCHEDULE THIS EO.DUTY.HOURS IN 24 HOURS                            00012510
  IF WEEKDAY.F(TIME.V) GT VWWDD                                      00012520
    RETURN **OFFICE IS CLOSED FOR THE WEEKEND                        00012530
  ELSE                                                               00012540
  CALL DISBAND.PARTIAL.TEAMS                                         00012550
  FOR EVERY WORKER CALLED WZX WITH WSTA(WZX) = IDLE DO               00012560
    LET WSTA(WZX) = OFF.DUTY                                         00012570
    SUBTRACT 1 FROM LNIW(WLEV(WZX))                                  00012580
  LOOP                                                               00012590
  FOR EVERY K IN EV.S(I.EO.ACTIVITY)                                 00012600
    WITH TIME.A(K)-TIME.V LT 30/(24*60) DO                           00012610
**  ACTIVITIES WITH LESS THAN 30 MINUTES REMAINING WILL BE COMPLETED 00012620
    LET MZD = EAMN(K)                                                00012630
    LET AZD = F.ROUTING(MZD)                                         00012640
      FOR EVERY WZD IN TEAM(AZD)                                     00012650
        LET WSTA(WZD) = OVERTIME                                     00012660
  LOOP                                                               00012670
  FOR EVERY K IN EV.S(I.EO.ACTIVITY)                                 00012680
    WITH TIME.A(K)-TIME.V GE 30/(24*60) AND                          00012690
         MNAP(EAMN(K)) NE 1 AND                                      00012700
         MNAP(EAMN(K)) NE 2 DO                                       00012710
    LET REASON = OFF.DUTY                                            00012720
    CALL INTERRUPT.AND.FILE GIVING K AND REASON                      00012730
  LOOP                                                               00012740
  FOR EVERY K IN EV.S(I.EO.ACTIVITY)                                 00012750
    WITH TIME.A(K)-TIME.V GE 30/(24*60) AND                          00012760
         MNAP(EAMN(K)) = 1 OR                                        00012770
         MNAP(EAMN(K)) = 2 DO                                        00012780
    LET MZD = EAMN(K)                                                00012790
    LET AZD = F.ROUTING(MZD)                                         00012800
      FOR EVERY WZD IN TEAM(AZD)                                     00012810
        LET WSTA(WZD) = OVERTIME                                     00012820
  LOOP                                                               00012830
```

```
CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                    00012840
CALL CHECK.QUEUE**REASSIGN WORKERS IF A PREEMPT OCCURRED          00012850
RETURN                                                            00012860
END                                                               00012870
```

```
..
     ROUTINE OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS                    00012880
                                                                          00012890
  CALL MY.TRACE(13)                                                       00012900
  DEFINE VAL AS REAL VARIABLE                                             00012910
  FOR EVERY MZF IN QUEUE WITH MNAP(MZF) = 1 AND                           00012920
    MTOA(MZF) + VRMPT GT TIME.V DO                                        00012930
    REMOVE MZF FROM QUEUE                                                 00012940
    FILE MZF IN WAIT.FOR.MISSION.TIME                                     00012950
      SCHEDULE AN ACCOMPLISH.PR1.MISSION AT MTOA(MZF) + VRMPT             00012960
  LOOP                                                                    00012970
  FOR EVERY MZF IN QUEUE WITH MNAP(MZF) = 1 OR                            00012980
                              MNAP(MZF) = 2 AND                           00012990
    ANWR(MZF) LE SUM.WORKERS(3,MZF,0) DO                                  00013000
    REMOVE THIS MZF FROM QUEUE                                            00013010
    IF ANWR(F.ROUTING(MZF)) = 0 **MANPOWER REQ IS EXTERNAL                00013020
      GO TO WORKER.FOUND                                                  00013030
    ELSE                                                                  00013040
    LET AZF = F.ROUTING(MZF)                                              00013050
    *PREEMPT.COMPLETED.LOOK.AGAIN*                                        00013060
    LET J=0                                                               00013070
    LET I = AFWL(AZF)                                                     00013080
    *LOOK.AGAIN*                                                          00013090
    IF I=0                                                                00013100
      GO TO V                                                             00013110
    ELSE                                                                  00013120
    *MORE.REQUIRED*                                                       00013130
    LET WZF = 0                                                           00013140
     FOR EVERY WORKER CALLED WZP WITH WLEV(WZP) = I AND                   00013150
      WSTA(WZP) NE OVERTIME                                               00013160
      COMPUTE WZF = MIN(WZP) OF TOBSERVED.STATE(WZP,5)                    00013170
    IF WZF NE 0                                                           00013180
      FILE WZF IN TEAM(AZF)                                               00013190
      LET WSTA(WZF) = OVERTIME                                            00013200
      SUBTRACT 1 FROM ANWR(AZF)                                           00013210
      IF ANWR(AZF) LT 0                                                   00013220
```

```
    CALL ERROR.PACKAGE(2,MZF)                                          00013230
  ALWAYS                                                               00013240
  IF ANWR(AZF) = 0                                                     00013250
    GO TO WORKER.FOUND                                                 00013260
  ELSE                                                                 00013270
  GO TO MORE.REQUIRED                                                  00013280
ELSE                                                                   00013290
ADD 1 TO J                                                             00013300
GO TO W, X, Y, V, U PER J                                              00013310
  "W" LET I = ASWL(AZF)                                                00013320
      GO TO LOOK.AGAIN                                                 00013330
  "X" LET I = ATWL(AZF)                                                00013340
      GO TO LOOK.AGAIN                                                 00013350
  "Y" IF MNAP(MZF) = 1                                                 00013360
        CALL PREEMPT                                                   00013370
        ADD 1 TO L                                                     00013380
        IF L GT N.WORKER                                               00013390
          GO TO V                                                      00013400
        ELSE                                                           00013410
        GO TO PREEMPT.COMPLETED.LOOK.AGAIN                             00013420
      ELSE                                                             00013430
      GO TO V                                                          00013440
  "V" UNTIL TEAM(AZF) IS EMPTY DO                                      00013450
        REMOVE THE FIRST MZF FROM TEAM(AZF)                            00013460
        ADD 1 TO ANWR(AZF)                                             00013470
        LET WSTA(MZF) = OFF.DUTY                                       00013480
      LOOP                                                             00013490
      FILE MZF IN QUEUE""ANOTHER PRIORITY ACTIVITY IN PROGRESS         00013500
      RETURN                                                           00013510
  "U" CALL ERROR.PACKAGE(3,MZF)                                        00013520
      "WORKER.FOUND"                                                   00013530
  LET VAL = VALUE(MZF)                                                 00013540
  IF WEEKDAY.F(TIME.V) GT VWWDD                                        00013550
    PRINT 6 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V),                 00013560
      YEAR.F(TIME.V), MTPN(MZF), MZF, ANOA(AZF), APOA(AZF),            00013570
```

```
        N.TEAM(AZF) AND VAL THUS                                          00013580
                                                                          00013590
                                                                          00013600
        A PRIORITY MISSION WAS INITIATED ON A NON DUTY DAY                00013610
                                                                          00013620
    DATE    TEST PHASE  MISSION  ACTIVITY  PR  WORKERS REQ  DURATION     00013630
** ** ****     ***       ******     ***    ***       ***     ****.* MIN  00013640
    ALWAYS                                                                00013650
  SCHEDULE AN EO.ACTIVITY GIVING MZF IN VAL MINUTES                       00013660
LOOP                                                                      00013670
RETURN                                                                    00013680
END                                                                       00013690
```

```
..                                                                    00013700
                    ROUTINE BUILD.PARTIAL.TEAMS                       00013710
  CALL MY.TRACE(20)                                                   00013720
  FOR EVERY MZF IN QUEUE WITH APAR(F.ROUTING(MZF)) = "YES" AND        00013730
    SUM.WORKERS(1,MZF,1) NE 0 DO                                      00013740
    REMOVE MZF FROM QUEUE                                             00013750
    CALL ALLOCATION(MZF,1)                                            00013760
  LOOP                                                                00013770
  RETURN                                                              00013780
  END                                                                 00013790
```

```
..
                ROUTINE DISBAND.PARTIAL.TEAMS                          00013800
CALL MY.TRACE(21)                                                      00013810
LET REASON = IDLE                                                      00013820
FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH                                00013830
  ANWR(F.ROUTING(EAMN(K))) NE 0                                        00013840
  CALL INTERRUPT.AND.FILE(K,REASON)                                    00013850
RETURN                                                                 00013860
END                                                                    00013870
                                                                       00013880
```

```
**                                                                    00013890
      ROUTINE TO COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION GIVEN MZG  00013900
  CALL MY.TRACE(14,MZG)                                             00013910
  IF MXCI(MZG) = 1                                                  00013920
    ADD 1 TO TCNX(MTPN(MZG))                                        00013930
    RETURN                                                          00013940
  ELSE                                                              00013950
  LET TSTY(MTPN(MZG)) = TIME.V - MTOA(MZG)                          00013960
  ADD MCEL(MZG) TO TCMP(MTPN(MZG))                                  00013970
  FOR I = 1 TO N.TEST.PHASE                                         00013980
  IF TCMP(I) GE TREQ(I)                                             00013990
      SCHEDULE AN EO.SIMULATION NOW                                 00014000
  ALWAYS                                                            00014010
  RETURN                                                            00014020
  END                                                               00014030
```

```
                EVENT WREPORT SAVING THE EVENT NOTICE                    00014040
                                                                         00014050
SCHEDULE THIS WREPORT IN 1 WK                                            00014060
CALL XWREPORT                                                            00014070
RETURN                                                                   00014080
END                                                                      00014090
```

```
..                                                                          00014100
                          ROUTINE XWREPORT                                  00014110
DEFINE P AS A REAL VARIABLE                                                 00014120
RESERVE VWCNF, VWCNX, VWCMP AS N.TEST.PHASE                                 00014130
START NEW PAGE                                                              00014140
PRINT 4 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V), YEAR.F(TIME.V) THUS      00014150
                          WEEKLY TEST FORCE REPORT                          00014160
                        (FOR WEEK ENDING **/**/****                         00014170
                                                                            00014180
                                                                            00014190
FOR EVERY TEST.PHASE DO                                                     00014200
  LET I = TEST.PHASE                                                        00014210
  LET J = TCNF(I) - VWCNF(I)                                                00014220
  LET K = TCNX(I) - VWCNX(I)                                                00014230
  LET P = TCMP(I) - VWCMP(I)                                                00014240
  IF REAL.F(J) = P = 0.0                                                    00014250
    GO LOOP                                                                 00014260
  ELSE                                                                      00014270
  PRINT 15 LINES WITH I, J, K, TREQ(I), P, TCMP(I), WAVG.STAY(I),           00014280
       WSTD.STAY(I), WSUM.STAY(I), AND WSOME.SQUARES.STAY(I) THUS           00014290
 MISSION STATISTICS                                                         00014300
 FOR TEST PHASE ** ARE                                                      00014310
                    NUMBER SCHEDULED---------- = ***      MISSIONS          00014320
                    NUMBER CANCELLED---------- = ***      MISSIONS          00014330
                    NUMBER REQUIRED----------- = ***      MISSIONS          00014340
                    EFFECTIVE PRODUCTIVITY---- = ***.**** MISSIONS          00014350
                    CUMULATIVE NUMBER COMPLETED- = ***.**** MISSIONS        00014360
                    AVERAGE MISSION DURATION---- = ***.**** DAYS            00014370
                    STD DEV------------------- = ***.**** DAYS            00014380
                    TOTAL OF DURATION TIMES---- = ***.**** DAYS            00014390
                    SUM OF SQ OF DURATION TIMES- = *****.** DAYS           00014400
                                                                            00014410
                                                                            00014420
                                                                            00014430
                                                                            00014440
```

```
  LET VWCNF(I) = TCNF(I)                                              00014450
  LET VWCNX(I) = TCNX(I)                                              00014460
  LET VWCMP(I) = TCMP(I)                                              00014470
  "LOOP"                                                              00014480
LOOP                                                                  00014490
PRINT 9 LINES WITH N.TEST.PHASE THUS                                  00014500
NOTE   STATISTICS ARE COMPUTED ONLY FOR MISSIONS                      00014510
       COMPLETED IN THIS REPORTING PERIOD.                            00014520
                                                                      00014530
                                                                      00014540
NOTE   IF DURING THIS REPORTING PERIOD NO MISSIONS WERE SCHEDULED OR  00014550
       COMPLETED IN A TEST PHASE, MISSION STATISTICS WERE OMITTED.    00014560
       THERE ARE *** TEST PHASES.                                     00014570
                                                                      00014580
                                                                      00014590
START NEW PAGE                                                        00014600
PRINT 3 LINES AS FOLLOWS                                              00014610
                              QUEUE REPORT                            00014620
                                                              SUM     00014630
MAX LENGTH     MIN LENGTH     AVG LENGTH     STD DEV     OF Q TIMES  00014640
PRINT 1 LINE WITH WQMAX.QUEUE, WQMIN.QUEUE, WQAVG.QUEUE,              00014650
  WQSTD.QUEUE, WQSUM.QUEUE THUS                                       00014660
***.* MSN      ***.* MSN      ***.** MSN ***.*** MSN    ****.* DAYS  00014670
SKIP 3 LINES                                                          00014680
BEGIN REPORT                                                          00014690
BEGIN HEADING                                                         00014700
PRINT 3 LINES THUS                                                    00014710
                    WEEKLY HISTOGRAM OF QUEUE LENGTH                  00014720
                                                                      00014730
QUEUE LENGTH      STATE PERCENTAGE                                    00014740
SKIP 1 LINE                                                           00014750
END""HEADING                                                          00014760
FOR I = INT.F(WQMIN.QUEUE + 1) TO INT.F(WQMAX.QUEUE + 1)              00014770
  PRINT 1 LINE WITH I-1 AND WFREQ(I)/.07 THUS                         00014780
*** MISSIONS            **.** %                                      00014790
```

```
END**REPORT                                                                    00014800
PRINT 5 LINES THUS                                                             00014810
                                                                               00014820
                                                                               00014830
NOTE   ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS.         00014840
       PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING              00014850
         MAY NOT SUM TO 100.                                                   00014860
SKIP 3 LINES                                                                   00014870
BEGIN REPORT ON A NEW PAGE PRINTING FOR LOFW = 1 TO N.LOFW IN                  00014880
  GROUPS OF 5                                                                  00014890
BEGIN HEADING                                                                  00014900
PRINT 6 LINES THUS                                                             00014910
             HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL                     00014920
                                                                               00014930
                                  STATE TIME                                   00014940
                                    (DAYS)                                     00014950
                                                                               00014960
NUMBER OF IDLE WORKERS  LEVEL 1  LEVEL 2  LEVEL 3  LEVEL 4  ...LEVEL N        00014970
END**HEADING                                                                   00014980
FOR I = 1 TO INT.F(LARGEST(1)) + 2 DO                                          00014990
PRINT 1 LINE WITH I-1 AND A GROUP OF                                           00015000
  WIDLE.WORKERS(LEVEL.OF.WORKER,I) FIELDS THUS                                 00015010
             ***      **.****  **.****  **.****  **.****  **.****              00015020
LOOP                                                                           00015030
FOR L = 1 TO N.LOFW DO                                                         00015040
  FOR EVERY WORKER WITH WLEV(WORKER) = L                                       00015050
    COMPUTE LTIT(L) = SUM OF WOBSERVED.STATE(WORKER,1)                         00015060
LOOP                                                                           00015070
PRINT 1 LINE THUS                                                              00015080
      TOTAL IDLE TIME                                                          00015090
PRINT 1 LINE WITH A GROUP OF LTIT(LEVEL.OF.WORKER) FIELDS THUS                 00015100
                        ****.**  ****.**  ****.**  ****.**  ****.**           00015110
END**REPORT                                                                    00015120
SKIP 3 LINES                                                                   00015130
BEGIN REPORT ON A NEW PAGE                                                     00015140
```

```
BEGIN HEADING                                                          00015150
PRINT 6 LINES THUS                                                     00015160
                          ANALYSIS OF WORKER STATUS                    00015170
                                                                       00015180
                                HOURS IN STATE                         00015190
                              STATE PERCENTAGES                        00015200
                                                                       00015210
WORKER        IDLE          BUSY          BREAK      OFF DUTY     OVERTIME  00015220
END**HEADING                                                           00015230
**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.            00015240
FOR EACH WORKER DO                                                     00015250
  PRINT 2 LINES WITH WNAM(WORKER),                                     00015260
                     WOBSERVED.STATE(WORKER,1) * 24,                   00015270
                     WOBSERVED.STATE(WORKER,2) * 24,                   00015280
                     WOBSERVED.STATE(WORKER,3) * 24,                   00015290
                     WOBSERVED.STATE(WORKER,4) * 24,                   00015300
                     WOBSERVED.STATE(WORKER,5) * 24,                   00015310
                     WOBSERVED.STATE(WORKER,1) / .07,                  00015320
                     WOBSERVED.STATE(WORKER,2) / .07,                  00015330
                     WOBSERVED.STATE(WORKER,3) / .07,                  00015340
                     WOBSERVED.STATE(WORKER,4) / .07,                  00015350
                     WOBSERVED.STATE(WORKER,5) / .07 THUS              00015360
******      **.** HRS   **.** HRS   **.** HRS   **.** HRS   **.** HRS  00015370
            **.** %     **.** %     **.** %     **.** %     **.** %    00015380
  SKIP 1 LINE                                                          00015390
LOOP                                                                   00015400
END**REPORT                                                            00015410
RESET WEEKLY TOTALS OF N.QUEUE                                         00015420
FOR EVERY LEVEL.OF.WORKER                                              00015430
  RESET WEEKLY TOTALS OF LNIW                                          00015440
FOR EVERY TEST.PHASE                                                   00015450
  RESET WEEKLY TOTALS OF TSTY                                          00015460
FOR EVERY WORKER                                                       00015470
  RESET WEEKLY TOTALS OF WSTA                                          00015480
RETURN                                                                 00015490
END                                                                    00015500
```

```
..
                EVENT MREPORT SAVING THE EVENT NOTICE                    00015510
SCHEDULE THIS MREPORT NEXT.MONTH                                         00015520
CALL XMREPORT                                                            00015530
RETURN                                                                   00015540
END                                                                      00015550
                                                                         00015560
```

```
                    ROUTINE XMREPORT                                          00015570
                                                                              00015580
DEFINE P AS A REAL VARIABLE                                                   00015590
RESERVE VMCNF, VMCNX, VMCMP AS N.TEST.PHASE                                   00015600
START NEW PAGE                                                                00015610
PRINT 3 LINES WITH                                                            00015620
  MONTH.F(TIME.V-8.) AND MOD.F(YEAR.F(TIME.V-8.),100) THUS                    00015630
                        REPORT FOR MONTH OF **/**                             00015640
                                                                              00015650
                                                                              00015660
FOR EVERY TEST.PHASE DO                                                       00015670
  LET I = TEST.PHASE                                                          00015680
  LET J = TCNF(I) - VMCNF(I)                                                  00015690
  LET K = TCNX(I) - VMCNX(I)                                                  00015700
  LET P = TCMP(I) - VMCMP(I)                                                  00015710
  IF REAL.F(J) = P = 0.0                                                      00015720
    GO LOOP                                                                   00015730
ELSE                                                                          00015740
  PRINT 15 LINES WITH I, J, K, TREQ(I), P, TCMP(I), MAVG.STAY(I),             00015750
       MSTD.STAY(I), MSUM.STAY(I), AND MSOME.SQUARES.STAY(I) THUS             00015760
 MISSION STATISTICS                                                           00015770
 FOR TEST PHASE ** ARE                                                        00015780
                    NUMBER SCHEDULED---------- = ***       MISSIONS           00015790
                    NUMBER CANCELLED---------- = ***       MISSIONS           00015800
                    NUMBER REQUIRED----------- = ***       MISSIONS           00015810
                    EFFECTIVE PRODUCTIVITY---- = ***.****  MISSIONS           00015820
                    CUMULATIVE NUMBER COMPLETED- = ***.****  MISSIONS         00015830
                    AVERAGE MISSION DURATION---- = ***.****  DAYS             00015840
                    STD DEV------------------- = ***.****  DAYS             00015850
                    TOTAL OF DURATION TIMES----- = ***.****  DAYS             00015860
                    SUM OF SQ OF DURATION TIMES- = *****.**  DAYS             00015870
                                                                              00015880
                                                                              00015890
                                                                              00015900
                                                                              00015910
```

```
  LET VMCNF(I) = TCNF(I)                                                    00015920
  LET VMCNX(I) = TCNX(I)                                                    00015930
  LET VMCMP(I) = TCMP(I)                                                    00015940
   'LOOP'                                                                   00015950
LOOP                                                                        00015960
PRINT 9 LINES WITH N.TEST.PHASE THUS                                        00015970
NOTE   STATISTICS ARE COMPUTED ONLY FOR MISSIONS                            00015980
       COMPLETED IN THIS REPORTING PERIOD.                                  00015990
                                                                            00016000
                                                                            00016010
NOTE   IF DURING THIS REPORTING PERIOD NO MISSIONS WERE SCHEDULED OR        00016020
       COMPLETED IN A TEST PHASE, MISSION STATISTICS WERE OMITTED.          00016030
       THERE ARE *** TEST PHASES.                                           00016040
                                                                            00016050
                                                                            00016060
START NEW PAGE                                                              00016070
PRINT 3 LINES AS FOLLOWS                                                    00016080
                                QUEUE REPORT                                00016090
                                                                     SUM    00016100
MAX LENGTH     MIN LENGTH     AVG LENGTH     STD DEV     OF Q TIMES       00016110
PRINT 1 LINE WITH MQMAX.QUEUE, MQMIN.QUEUE, MQAVG.QUEUE,                    00016120
  MQSTD.QUEUE, MQSUM.QUEUE THUS                                             00016130
***.* MSN      ***.* MSN      ***.** MSN ***.*** MSN    ****.* DAYS        00016140
SKIP 3 LINES                                                                00016150
BEGIN REPORT                                                                00016160
BEGIN HEADING                                                               00016170
PRINT 3 LINES THUS                                                          00016180
                      MONTHLY HISTOGRAM OF QUEUE LENGTH                     00016190
                                                                            00016200
QUEUE LENGTH      STATE TIME                                                00016210
SKIP 1 LINE                                                                 00016220
END''HEADING                                                                00016230
FOR I = INT.F(MQMIN.QUEUE + 1) TO INT.F(MQMAX.QUEUE + 1)                    00016240
  PRINT 1 LINE WITH I-1 AND MFREQ(I) THUS                                   00016250
*** MISSIONS            **.**** DAYS                                       00016260
```

```
END**REPORT                                                              00016270
PRINT 3 LINE THUS                                                        00016280
                                                                         00016290
                                                                         00016300
NOTE   ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS    00016310
SKIP 3 LINES                                                             00016320
BEGIN REPORT ON A NEW PAGE PRINTING FOR LOFW = 1 TO N.LOFW IN            00016330
  GROUPS OF 5                                                            00016340
BEGIN HEADING                                                            00016350
PRINT 6 LINES THUS                                                       00016360
              HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL              00016370
                                                                         00016380
                                  STATE TIME                             00016390
                                    (DAYS)                               00016400
                                                                         00016410
NUMBER OF IDLE WORKERS  LEVEL 1  LEVEL 2  LEVEL 3  LEVEL 4  ...LEVEL N  00016420
END**HEADING                                                             00016430
FOR I = 1 TO INT.F(LARGEST(2)) + 2 DO                                    00016440
  PRINT 1 LINE WITH I-1 AND A GROUP OF                                   00016450
    MIDLE.WORKERS(LEVEL.OF.WORKER,I) FIELDS THUS                         00016460
            ***     **.****  **.****  **.****  **.****  **.****         00016470
LOOP                                                                     00016480
FOR L = 1 TO N.LOFW DO                                                   00016490
  FOR EVERY WORKER WITH WLEV(WORKER) = L                                 00016500
    COMPUTE LTIT(L) = SUM OF MOBSERVED.STATE(WORKER,1)                   00016510
LOOP                                                                     00016520
PRINT 1 LINE THUS                                                        00016530
      TOTAL IDLE TIME                                                    00016540
PRINT 1 LINE WITH A GROUP OF LTIT(LEVEL.OF.WORKER) FIELDS THUS           00016550
                      ****.**  ****.**  ****.**  ****.**  ****.**       00016560
END**REPORT                                                              00016570
SKIP 3 LINES                                                             00016580
BEGIN REPORT ON A NEW PAGE                                               00016590
BEGIN HEADING                                                            00016600
PRINT 6 LINES THUS                                                       00016610
```

```
                                ANALYSIS OF WORKER STATUS                                    00016620
                                                                                             00016630
                                          STATE TIME                                         00016640
                                           (HOURS)                                           00016650
                                                                                             00016660
  WORKER        IDLE          BUSY          BREAK        OFF DUTY      OVERTIME              00016670
  END**HEADING                                                                               00016680
**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.                                   00016690
  FOR EACH WORKER DO                                                                         00016700
    PRINT 1 LINES WITH WNAM(WORKER),                                                         00016710
                      MOBSERVED.STATE(WORKER,1) * 24,                                        00016720
                      MOBSERVED.STATE(WORKER,2) * 24,                                        00016730
                      MOBSERVED.STATE(WORKER,3) * 24,                                        00016740
                      MOBSERVED.STATE(WORKER,4) * 24,                                        00016750
                      MOBSERVED.STATE(WORKER,5) * 24 THUS                                    00016760
  ******    **.** HRS    **.** HRS    **.** HRS    **.** HRS    **.** HRS                    00016770
  LOOP                                                                                       00016780
  END**REPORT                                                                                00016790
  RESET MONTHLY TOTALS OF N.QUEUE                                                            00016800
  FOR EVERY LEVEL.OF.WORKER                                                                  00016810
    RESET MONTHLY TOTALS OF LNIW                                                             00016820
  FOR EVERY TEST.PHASE                                                                       00016830
    RESET MONTHLY TOTALS OF TSTY                                                             00016840
  FOR EVERY WORKER                                                                           00016850
    RESET MONTHLY TOTALS OF WSTA                                                             00016860
  RETURN                                                                                     00016870
  END                                                                                        00016880
```

```
..                                                                      00016890
                    EVENT FINAL.REPORT                                  00016900
START NEW PAGE                                                          00016910
PRINT 4 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V), YEAR.F(TIME.V) THUS  00016920
                         FINAL TEST FORCE REPORT                        00016930
                   (FOR SIMULATION ENDING ** ** ****)                   00016940
                                                                        00016950
                                                                        00016960
FOR EVERY TEST.PHASE DO                                                 00016970
  LET I = TEST.PHASE                                                    00016980
  PRINT 16 LINES WITH I, TCNF(I), TCNX(I), TREQ(I), TCMP(I),            00016990
    TAVG.STAY(I), TSTD.STAY(I), TSUM.STAY(I), TSOME.SQUARES.STAY(I),    00017000
    MONTH.F(TDCD(I)), DAY.F(TDCD(I)), AND YEAR.F(TDCD(I)) THUS          00017010
 MISSION STATISTICS                                                     00017020
 FOR TEST PHASE ** ARE                                                  00017030
                      NUMBER SCHEDULED------------ = ***      MISSIONS  00017040
                      NUMBER CANCELLED------------ = ***      MISSIONS  00017050
                      NUMBER REQUIRED------------- = ***      MISSIONS  00017060
                      EFFECTIVE PRODUCTIVITY------ = ***.**** MISSIONS  00017070
                      AVERAGE MISSION DURATION---- = ***.**** DAYS      00017080
                      STD DEV--------------------- = ***.**** DAYS      00017090
                      TOTAL OF DURATION TIMES----- = ***.**** DAYS      00017100
                      SUM OF SQ OF DURATION TIMES- = *****.** DAYS      00017110
                      DATE COMPLETED-------------- = ** ** ****         00017120
                                                                        00017130
                                                                        00017140
                                                                        00017150
                                                                        00017160
                                                                        00017170
LOOP                                                                    00017180
PRINT 5 LINES THUS                                                      00017190
NOTE    SINCE THE DURATION OF CANCELLED MISSIONS IS ZERO,               00017200
        STATISTICS ARE COMPUTED FOR COMPLETED MISSIONS ONLY.            00017210
                                                                        00017220
                                                                        00017230
```

```
START NEW PAGE                                                          00017240
PRINT 3 LINES AS FOLLOWS                                                00017250
                                QUEUE REPORT                            00017260
                                                                   SUM  00017270
 MAX LENGTH          AVG LENGTH          STD DEV              OF Q TIMES00017280
PRINT 1 LINE WITH TQMAX.QUEUE,TQAVG.QUEUE,TQSTD.QUEUE,TQSUM.QUEUE THUS 00017290
 ***.* MISSIONS      ***.**** MISSIONS   ***.**** MISSIONS  ***.* DAYS 00017300
SKIP 3 LINES                                                            00017310
BEGIN REPORT                                                            00017320
BEGIN HEADING                                                           00017330
PRINT 3 LINES THUS                                                      00017340
                          HISTOGRAM OF QUEUE LENGTH                     00017350
                                                                        00017360
QUEUE LENGTH      STATE PERCENTAGE                                      00017370
SKIP 1 LINE                                                             00017380
END**HEADING                                                            00017390
FOR I = 1 TO INT.F(TQMAX.QUEUE + 1)                                     00017400
  PRINT 1 LINE WITH I-1 AND (TFREQ(I)/TIME.V)*100 THUS                  00017410
*** MISSIONS           **.** %                                          00017420
END**REPORT                                                             00017430
PRINT 5 LINES THUS                                                      00017440
                                                                        00017450
                                                                        00017460
NOTE   ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS.  00017470
       PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING       00017480
         MAY NOT SUM TO 100.                                            00017490
SKIP 3 LINES                                                            00017500
BEGIN REPORT ON A NEW PAGE PRINTING FOR LOFW = 1 TO N.LOFW IN           00017510
  GROUPS OF 5                                                           00017520
BEGIN HEADING                                                           00017530
PRINT 6 LINES THUS                                                      00017540
              HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL             00017550
                                                                        00017560
                          STATE TIME                                    00017570
                                                                        00017580
```

```
                                    (DAYS)                                      00017590
                                                                                00017600
NUMBER OF IDLE WORKERS  LEVEL 1  LEVEL 2  LEVEL 3  LEVEL 4  ...LEVEL N00017610
END**HEADING                                                                    00017620
FOR I = 1 TO INT.F(LARGEST(3)) + 2 DO                                           00017630
  PRINT 1 LINE WITH I-1 AND A GROUP OF                                          00017640
    TIDLE.WORKERS(LEVEL.OF.WORKER,I) FIELDS THUS                                00017650
              ***      **.****  **.****  **.****  **.****  **.****             00017660
LOOP                                                                            00017670
FOR L = 1 TO N.LOFW DO                                                          00017680
  FOR EVERY WORKER WITH WLEV(WORKER) = L                                        00017690
    COMPUTE LTIT(L) = SUM OF TOBSERVED.STATE(WORKER,1)                          00017700
LOOP                                                                            00017710
PRINT 1 LINE THUS                                                               00017720
       TOTAL IDLE TIME                                                          00017730
PRINT 1 LINE WITH A GROUP OF LTIT(LEVEL.OF.WORKER) FIELDS THUS                  00017740
                      ****.**  ****.**  ****.**  ****.**  ****.**              00017750
END**REPORT                                                                     00017760
SKIP 3 LINES                                                                    00017770
BEGIN REPORT ON A NEW PAGE                                                      00017780
BEGIN HEADING                                                                   00017790
PRINT 6 LINES THUS                                                              00017800
                         ANALYSIS OF WORKER STATUS                              00017810
                                                                                00017820
                              HOURS IN STATE                                    00017830
                            STATE PERCENTAGES                                   00017840
                                                                                00017850
WORKER       IDLE       BUSY       BREAK    OFF DUTY    OVERTIME               00017860
END**HEADING                                                                    00017870
**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.                     00017880
FOR EACH WORKER DO                                                              00017890
  PRINT 2 LINES WITH WNAM(WORKER),                                              00017900
                    TOBSERVED.STATE(WORKER,1) * 24,                             00017910
                    TOBSERVED.STATE(WORKER,2) * 24,                             00017920
                    TOBSERVED.STATE(WORKER,3) * 24,                             00017930
```

```
                    TOBSERVED.STATE(WORKER,4) * 24,                                  00017940
                    TOBSERVED.STATE(WORKER,5) * 24,                                  00017950
                    (TOBSERVED.STATE(WORKER,1)/TIME.V)*100,                          00017960
                    (TOBSERVED.STATE(WORKER,2)/TIME.V)*100,                          00017970
                    (TOBSERVED.STATE(WORKER,3)/TIME.V)*100,                          00017980
                    (TOBSERVED.STATE(WORKER,4)/TIME.V)*100,                          00017990
                    (TOBSERVED.STATE(WORKER,5)/TIME.V)*100 THUS                      00018000
******      **.** HRS    **.** HRS    **.** HRS    **.** HRS    **.** HRS            00018010
          **.** %      **.** %      **.** %      **.** %      **.** %                00018020
  SKIP 1 LINE                                                                        00018030
LOOP                                                                                 00018040
END**REPORT                                                                          00018050
CALL CLEAR                                                                           00018060
RETURN                                                                               00018070
END                                                                                  00018080
```

```
..                                                                    00018090
                        ROUTINE LARGEST GIVEN FUNCT                   00018100
DEFINE X AS A REAL VARIABLE                                           00018110
GO TO W, M, T PER FUNCT                                               00018120
  "W"  FOR EACH LEVEL.OF.WORKER                                       00018130
         COMPUTE X AS THE MAXIMUM OF WLMAX.IDLE.WORKERS(LOFW)         00018140
       JUMP AHEAD                                                     00018150
  "M"  FOR EACH LEVEL.OF.WORKER                                       00018160
         COMPUTE X AS THE MAXIMUM OF MLMAX.IDLE.WORKERS(LOFW)         00018170
       JUMP AHEAD                                                     00018180
  "T"  FOR EACH LEVEL.OF.WORKER                                       00018190
         COMPUTE X AS THE MAXIMUM OF TLMAX.IDLE.WORKERS(LOFW)         00018200
       HERE                                                           00018210
       RETURN WITH X                                                  00018220
END                                                                   00018230
```

```
..                                                                          00018240
                ROUTINE MY.TRACE(NUMB,IDENT)                                00018250
  IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                                  00018260
  GO TO 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 PER NUMB      00018270
 *1* WRITE TIME.V AS                                                        00018280
      "CHECK.QUEUE WAS CALLED AT", D(10,4),/                                00018290
     JUMP AHEAD                                                             00018300
 *2* WRITE IDENT AND TIME.V AS                                              00018310
       "A BRK TIME ACTIVITY IN MSN", I 6, " WAS STARTED AT", D(10,4),/      00018320
     JUMP AHEAD                                                             00018330
 *3* WRITE IDENT AND TIME.V AS                                              00018340
      "DET PREMSN CNX FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/        00018350
     JUMP AHEAD                                                             00018360
 *4* WRITE IDENT AND TIME.V AS                                              00018370
      "CAL MSN EFF FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/           00018380
     JUMP AHEAD                                                             00018390
 *5* WRITE IDENT AND TIME.V AS                                              00018400
      "ATTEND.TO.MISSION FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/     00018410
     JUMP AHEAD                                                             00018420
 *6* WRITE IDENT AND TIME.V AS                                              00018430
      "SUM.WORKERS FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/           00018440
     JUMP AHEAD                                                             00018450
 *7* WRITE IDENT AND TIME.V AS                                              00018460
     "ALLOCATION FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/             00018470
     JUMP AHEAD                                                             00018480
 *8* WRITE IDENT AND TIME.V AS                                              00018490
      "VAL NUM WRKS REQ FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/      00018500
     JUMP AHEAD                                                             00018510
 *9* WRITE TIME.V AS                                                        00018520
      "PREEMPT WAS CALLED AT", D(10,4),/                                    00018530
     JUMP AHEAD                                                             00018540
 *10* WRITE IDENT AND TIME.V AS                                             00018550
      "INTERRUPT.AND.FILE FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/    00018560
      JUMP AHEAD                                                            00018570
 *11* WRITE IDENT AND TIME.V AS                                             00018580
```

```
      "VALUE FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/                00018590
      JUMP AHEAD                                                           00018600
*12*  WRITE IDENT AND TIME.V AS                                            00018610
      "LACM FOR TEST PHASE", I 3, " WAS CALLED AT", D(10,4),/              00018620
      JUMP AHEAD                                                           00018630
*13*  WRITE TIME.V AS                                                      00018640
      "OVERTIME.ACTIVITIES.FOR.PRIORITY.MSN WAS CALLED AT", D(10,4),/      00018650
      JUMP AHEAD                                                           00018660
*14*  WRITE IDENT AND TIME.V AS                                            00018670
      "CSTARC FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/               00018680
      JUMP AHEAD                                                           00018690
*15*  WRITE IDENT AND TIME.V AS                                            00018700
      "QENTER.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/          00018710
      JUMP AHEAD                                                           00018720
*16*  WRITE IDENT AND TIME.V AS                                            00018730
       "QEXIT.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/          00018740
      JUMP AHEAD                                                           00018750
*17*  WRITE IDENT AND TIME.V AS                                            00018760
       "WENTER.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/         00018770
      JUMP AHEAD                                                           00018780
*18*  WRITE IDENT AND TIME.V AS                                            00018790
       "WEXIT.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/          00018800
      JUMP AHEAD                                                           00018810
*19*  WRITE IDENT AND TIME.V AS                                            00018820
       "MSN.SCHEDULE FOR TEST PHASE", I 3, " WAS CALLED AT",D(10,4),/      00018830
      JUMP AHEAD                                                           00018840
*20*  WRITE TIME.V AS                                                      00018850
        "BUILD.PARTIAL.TEAMS WAS CALLED AT", D(10,4),/                     00018860
      JUMP AHEAD                                                           00018870
*21*  WRITE TIME.V AS                                                      00018880
        "DISBAND.PARTIAL.TEAMS WAS CALLED AT", D(10,4),/                   00018890
      HERE                                                                 00018900
ALWAYS                                                                     00018910
RETURN                                                                     00018920
END                                                                        00018930
```

```
**                                                                              00018940
      ROUTINE MYE.TRACE                                                         00018950
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                                        00018960
PRINT 3 LINES THUS                                                              00018970
                                                                                00018980
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00018990
                                                                                00019000
SKIP 1 LINE                                                                     00019010
GO TO EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,                                00019020
  CONFIRMED.MISSION,TAKE.BREAK,ACCOMPLISH.PR1.MISSION,                          00019030
  BEGIN.WORKING, WREPORT, MREPORT OR FINAL.REPORT PER EVENT.V                   00019040
*EO.ACTIVITY*                                                                   00019050
  LET MZH = EAMN(EO.ACTIVITY)                                                   00019060
  LET AZH = F.ROUTING(MZH)                                                      00019070
  PRINT 1 LINE WITH ANOA(AZH), MZH, AND TIME.V THUS                             00019080
  ACTIVITY  NUMBER ** OF MSN NUMBER ****** WAS COMPLETED AT ***.****            00019090
CALL MY.LIST(5,MZH)**ACTIVITIES IN ROUTING(MZH)                                 00019100
  JUMP AHEAD                                                                    00019110
*EO.DUTY.HOURS*                                                                 00019120
  WRITE TIME.V AS "DUTY HOURS ENDED AT", D(10,4),/                              00019130
  JUMP AHEAD                                                                    00019140
*EO.SIMULATION*                                                                 00019150
  WRITE TIME.V AS "SIMULATION ENDED AT", D(10,4),/                              00019160
  JUMP AHEAD                                                                    00019170
*CONFIRMED.MISSION*                                                             00019180
  PRINT 1 LINE WITH CTPN(CONFIRMED.MISSION) AND TIME.V THUS                     00019190
  A CNF MSN IN TEST PHASE ** WAS RECEIVED AT ******.****                        00019200
  JUMP AHEAD                                                                    00019210
*TAKE.BREAK*                                                                    00019220
  WRITE TIME.V AS "TAKE BREAK AT", D(10,4),/                                    00019230
  JUMP AHEAD                                                                    00019240
*ACCOMPLISH.PR1.MISSION*                                                        00019250
  WRITE TIME.V AS "A PRIORITY 1 ACTIVITY WAS INITIATED AT", D(10,4),/           00019260
  RETURN                                                                        00019270
*BEGIN.WORKING*                                                                 00019280
```

```
  WRITE TIME.V AS "BEGIN WORKING AT", D(10,4),/                    00019290
  JUMP AHEAD                                                       00019300
*WREPORT*                                                          00019310
  WRITE TIME.V AS "A WEEKLY REPORT WAS GENERATED AT", D(10,4),/    00019320
  GO WAITING.INFO                                                  00019330
*MREPORT*                                                          00019340
  WRITE TIME.V AS "A MONTHLY REPORT WAS GENERATED AT", D(10,4),/   00019350
  GO WAITING.INFO                                                  00019360
*FINAL.REPORT*                                                     00019370
  WRITE TIME.V AS "THE FINAL REPORT WAS GENERATED AT", D(10,4),/   00019380
  *WAITING.INFO*                                                   00019390
  CALL MY.LIST(6)**MISSIONS IN QUEUE                               00019400
  CALL MY.LIST(7)**MISSIONS IN WAIT FOR MISSION TIME               00019410
    HERE                                                           00019420
  IF WEEKDAY.F(TIME.V) LE VWWDD                                    00019430
    CALL MY.LIST(1)**WORKERS                                       00019440
  ALWAYS                                                           00019450
  SKIP 3 LINES                                                     00019460
  ALWAYS                                                           00019470
  RETURN                                                           00019480
END                                                                00019490
```

```
..                                                                    00019500
                    ROUTINE MY.LIST(NUMB,ID)                          00019510
  GO TO 1,2,3,4,5,6,7,8,9,10,11,12 PER NUMB                           00019520
  *1* PRINT 5 LINES THUS                                              00019530
                                                                      00019540
                                                                      00019550
      ATTRIBUTES OF EACH WORKER                                       00019560
                                                                      00019570
      WORKER     WNAM     WLEV     WSTA     M.TEAM                   00019580
    FOR EACH WORKER PRINT 1 LINE WITH WORKER, WNAM, WLEV, WSTA,       00019590
      M.TEAM THUS                                                     00019600
       ***       ******   ***        *        *                     00019610
    JUMP AHEAD                                                        00019620
  *2* PRINT 5 LINES THUS                                              00019630
                                                                      00019640
                                                                      00019650
      ATTRIBUTES OF EACH LEVEL OF WORKER                              00019660
                                                                      00019670
      LEVEL    LNIW                                                   00019680
    FOR EACH LEVEL.OF.WORKER PRINT 1 LINE WITH LOFW, LNIW(LOFW) THUS  00019690
      ***      ***                                                    00019700
    JUMP AHEAD                                                        00019710
  *3* PRINT 5 LINES THUS                                              00019720
                                                                      00019730
                                                                      00019740
    ATTRIBUTES OF TEST PHASE                                          00019750
                                                                      00019760
  TNUM   TREQ   TCNF   TCMP   TCNX   TDCD   TDBM   N.STRUC   N.CX.PROB 00019770
    PRINT 1 LINE WITH TNUM(ID), TREQ(ID), TCNF(ID), TCMP(ID),         00019780
      TCNX(ID), TDCD(ID), TDBM(ID), N.STRUCTURE(ID),                  00019790
      N.TCANCELLATION.PROBABILITY(ID) THUS                            00019800
  ***  ***   ***   ***.*  ***   ***.*  **.**     ***        ***      00019810
    JUMP AHEAD                                                        00019820
  *4* PRINT 5 LINES THUS                                              00019830
                                                                      00019840
```

```
                                                                        00019850
      ATTRIBUTES OF EACH ACTIVITY IN STRUCTURE(TEST PHASE)              00019860
                                                                        00019870
 ATPC   ANOA   APOA   ATFC   AICT   ANWR   APAR   AFWL   ASWL   ATWL   00019880
     FOR EVERY AZG IN STRUCTURE(ID) PRINT 1 LINE WITH ATPC(AZG),        00019890
       ANOA(AZG), APOA(AZG), ATFC(AZG), AICT(AZG), ANWR(AZG),           00019900
       APAR(AZG), AFWL(AZG), ASWL(AZG), ATWL(AZG) THUS                  00019910
 ****   ***    ***    ***    **.*   ***    ***    ***    ***    ***    00019920
     PRINT 2 LINES THUS                                                 00019930
                                                                        00019940
 ATPC    ACEL    APMA    ALRL    AURL    M.STRUCTURE    N.TEAM         00019950
     FOR EVERY AZG IN STRUCTURE(ID) PRINT 1 LINE WITH ATPC(AZG),        00019960
       ACEL(AZG), APMA(AZG), ALRL(AZG), AURL(AZG), M.STRUCTURE(AZG)     00019970
       AND N.TEAM(AZG) THUS                                             00019980
 ****   **.**   **.**   **.**   **.**       *          ***               00019990
     JUMP AHEAD                                                         00020000
 *5* PRINT 4 LINES THUS                                                 00020010
                                                                        00020020
                                                                        00020030
      ATTRIBUTES OF EACH ACTIVITY IN ROUTING(MISSION)                   00020040
                                                                        00020050
     PRINT 1 DOUBLE LINE THUS                                           00020060
 ATPC   ANOA   APOA   ATFC   AICT   ANWR   APAR   AFWL   ASWL   ATWL   00020070
ACEL   APMA   ALRL   AURL   M.ROUTING   N.TEAM                          00020080
     FOR EACH AZG IN ROUTING(ID) PRINT 2 DOUBLE LINE WITH ATPC(AZG),    00020090
       ANOA(AZG),APOA(AZG),ATFC(AZG),AICT(AZG),ANWR(AZG),APAR(AZG),     00020100
       AFWL(AZG), ASWL(AZG), ATWL(AZG), ACEL(AZG), APMA(AZG),           00020110
       ALRL(AZG), AURL(AZG), M.ROUTING(AZG) AND N.TEAM(AZG) THUS        00020120
 ****   ***    ***    ***    *.**** ***    ***    ***    ***    ***    00020130
*.**   **.**  **.**  **.**       *          ***                         00020140
     JUMP AHEAD                                                         00020150
 *6* IF QUEUE IS NOT EMPTY PRINT 4 LINES THUS                           00020160
                                                                        00020170
                                                                        00020180
      ATTRIBUTES OF EACH MISSION IN QUEUE                               00020190
```

```
                                                                        00020200
      PRINT 1 DOUBLE LINE THUS                                          00020210
MISSION     MTPN     MCEL     MTOA     MNAC     MNAP     MXCI     M.QUEUE  00020220
M.WFMT     N.CX.PROB     N.ROUTING                                      00020230
      FOR EVERY MZP IN QUEUE PRINT 1 DOUBLE LINE WITH MZP, MTPN(MZP),   00020240
        MCEL(MZP), MTOA(MZP), MNAC(MZP), MNAP(MZP), MXCI(MZP),          00020250
        M.QUEUE(MZP), M.WAIT.FOR.MISSION.TIME(MZP),                     00020260
        N.MCANCELLATION.PROBABILITY(MZP) AND N.ROUTING(MZP) THUS        00020270
******      ***    **.**** ***.****  ***      ***       *        *      00020280
   *          ***             ***                                       00020290
    ALWAYS                                                              00020300
    JUMP AHEAD                                                          00020310
*7* IF WAIT.FOR.MISSION.TIME IS NOT EMPTY PRINT 4 LINES THUS            00020320
                                                                        00020330
                                                                        00020340
      ATTRIBUTES OF EACH MISSION IN WAIT FOR MISSION TIME               00020350
                                                                        00020360
      PRINT 1 DOUBLE LINE THUS                                          00020370
MISSION     MTPN     MCEL     MTOA     MNAC     MNAP     MXCI     M.QUEUE  00020380
M.WFMT     N.CX.PROB     N.ROUTING                                      00020390
      DEFINE WFMT TO MEAN WAIT.FOR.MISSION.TIME                         00020400
      FOR EVERY MZP IN WFMT PRINT 1 DOUBLE LINE WITH MZP, MTPN(MZP),    00020410
        MCEL(MZP), MTOA(MZP), MNAC(MZP), MNAP(MZP), MXCI(MZP),          00020420
        M.QUEUE(MZP), M.WAIT.FOR.MISSION.TIME(MZP),                     00020430
        N.MCANCELLATION.PROBABILITY(MZP) AND N.ROUTING(MZP) THUS        00020440
******      ***    **.**** ***.****  ***      ***       *        *      00020450
   *          ***             ***                                       00020460
    ALWAYS                                                              00020470
    JUMP AHEAD                                                          00020480
*8* IF TCANCELLATION.PROBABILITY IS NOT EMPTY PRINT 5 LINES THUS        00020490
                                                                        00020500
                                                                        00020510
      ATTRIBUTES OF EACH XFACTOR IN TCANCELLATION.PROB(TEST PHASE)      00020520
                                                                        00020530
      XUPL     M.TCANCELLATION.PROBABILITY                              00020540
```

```
      FOR EVERY FCT IN TCANCELLATION.PROBABILITY(ID) PRINT 1 LINE      00020550
        WITH XUPL(FCT) AND M.TCANCELLATION.PROBABILITY(FCT) THUS        00020560
     **.**                    *                                         00020570
     ALWAYS                                                            00020580
   JUMP AHEAD                                                          00020590
 *9* BEGIN REPORT PRINTING FOR TEST.PHASE = 1 TO N.TEST.PHASE IN       00020600
       GROUPS OF 8                                                     00020610
     BEGIN HEADING                                                     00020620
     PRINT 7 LINES THUS                                                00020630
                                                                       00020640
                                                                       00020650
     1ARG FOR EACH ACTIVITY FOR EACH TEST PHASE                        00020660
                                                                       00020670
                                  TEST PHASE                           00020680
          1       2       3       4       5       6       7       N   00020690
   ACT                                                                 00020700
     END**HEADING                                                      00020710
     FOR I = 1 TO N.KIND.OF.ACTIVITY                                   00020720
      PRINT 1 LINE WITH I AND A GROUP OF 1ARG(TEST.PHASE,I) FIELDS THUS00020730
   ***  *****.* *****.* *****.* *****.* *****.* *****.* *****.* *****.*00020740
     END**REPORT                                                       00020750
     JUMP AHEAD                                                        00020760
 *10*BEGIN REPORT PRINTING FOR TEST.PHASE = 1 TO N.TEST.PHASE IN       00020770
       GROUPS OF 8                                                     00020780
     BEGIN HEADING                                                     00020790
     PRINT 7 LINES THUS                                                00020800
                                                                       00020810
                                                                       00020820
     2ARG FOR EACH ACTIVITY FOR EACH TEST PHASE                        00020830
                                                                       00020840
                                  TEST PHASE                           00020850
          1       2       3       4       5       6       7       N   00020860
   ACT                                                                 00020870
     END**HEADING                                                      00020880
     FOR I = 1 TO N.KIND.OF.ACTIVITY                                   00020890
```

```
    PRINT 1 LINE WITH I AND A GROUP OF 2ARG(TEST.PHASE,I) FIELDS THUS00020900
***  *****.* *****.* *****.* *****.* *****.* *****.* *****.* *****.*00020910
  END**REPORT                                                        00020920
  JUMP AHEAD                                                         00020930
*11*BEGIN REPORT PRINTING FOR WORKER = 1 TO N.WORKER IN GROUPS OF 13  00020940
  BEGIN HEADING                                                      00020950
  PRINT 7 LINES THUS                                                 00020960
                                                                     00020970
                                                                     00020980
    THE SKILL FACTOR OF EACH WORKER IN PERFORMING AN ACTIVITY IS      00020990
                                                                     00021000
                          WORKER                                     00021010
      1   2   3   4   5   6   7   8   9   10  11  12   N           00021020
ACT                                                                  00021030
  END**HEADING                                                       00021040
  FOR I = 1 TO N.KIND.OF.ACTIVITY                                    00021050
    PRINT 1 LINE WITH I AND A GROUP OF SKILL.FACTOR(WORKER,I)        00021060
      FIELDS THUS                                                    00021070
*** *.** *.** *.** *.** *.** *.** *.** *.** *.** *.** *.** *.** *.** 00021080
    END**REPORT                                                      00021090
    JUMP AHEAD                                                       00021100
*12*PRINT 5 LINES THUS                                              00021110
                                                                     00021120
                                                                     00021130
ATTRIBUTES OF EACH KIND OF ACTIVITY                                  00021140
                                                                     00021150
ACT  KNAM   KPOA  KTFC  KPAR  KFWL  KSWL  KTWL  KPMA  KLRL  KURL    00021160
FOR EVERY KIND.OF.ACTIVITY PRINT 1 LINE WITH KOA, KNAM(KOA),         00021170
  KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA), KSWL(KOA), KTWL(KOA),  00021180
  KPMA(KOA), KLRL(KOA), KURL(KOA) THUS                               00021190
***  ****** ***   ***   ***  ***   ***   ***   **.** **.** **.**    00021200
    HERE                                                             00021210
  SKIP 1 LINE                                                        00021220
  RETURN                                                             00021230
  END                                                                00021240
```

```
                ROUTINE QENTER.INFO GIVEN MZN                    00021250
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                         00021260
  CALL MY.TRACE(15,MZN)                                          00021270
  CALL INFO.PACKAGE(MZN)                                         00021280
  ALWAYS                                                         00021290
RETURN                                                           00021300
END                                                              00021310
                                                                 00021320
```

```
..                                                                  00021330
              ROUTINE QEXIT.INFO GIVEN MZN                          00021340
  IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                          00021350
    CALL MY.TRACE(16,MZN)                                           00021360
    CALL INFO.PACKAGE(MZN)                                          00021370
    ALWAYS                                                          00021380
  RETURN                                                            00021390
  END                                                               00021400
```

```
..
                ROUTINE WENTER.INFO GIVEN MZN                    00021410
  IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                       00021420
    CALL MY.TRACE(17,MZN)                                        00021430
    CALL INFO.PACKAGE(MZN)                                       00021440
    ALWAYS                                                       00021450
  RETURN                                                         00021460
  END                                                            00021470
                                                                 00021480
```

```
..
                ROUTINE WEXIT.INFO GIVEN MZN                          00021490
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX                              00021500
  CALL MY.TRACE(18,MZN)                                               00021510
  CALL INFO.PACKAGE(MZN)                                              00021520
  ALWAYS                                                              00021530
RETURN                                                                00021540
END                                                                   00021550
                                                                      00021560
```

```
**
                ROUTINE ERROR.PACKAGE(NUMB,ID)                                  00021570
                                                                                00021580
    GO TO 1,2,3,4 PER NUMB                                                      00021590
      *1* PRINT 9 LINES WITH READ.V, RECORD.V(READ.V), RCOLUMN.V THUS            00021600
                                                                                00021610
                                                                                00021620
          ERROR - EITHER TOO MUCH DATA OR DATA WAS INCORRECTLY READ.            00021630
                                                                                00021640
                                                                                00021650
                              INPUT STATUS                                      00021660
                                                                                00021670
                     UNIT      CARD NUMBER      COLUMN                        00021680
                      **          ****            **                          00021690
          STOP                                                                  00021700
      *2* PRINT 4 LINES WITH ATPC(F.ROUTING(ID)), ID, AND TIME.V THUS            00021710
                                                                                00021720
                                                                                00021730
          ERROR - NUMBER OF WORKERS REQ FOR ACTIVITY CODE *** IN                00021740
                  MISSION ****** IS NEGATIVE. TRACE ABOUT DAY ****.****         00021750
          JUMP AHEAD                                                            00021760
      *3* PRINT 4 LINES WITH ATPC(F.ROUTING(ID)) AND TIME.V THUS                 00021770
                                                                                00021780
                                                                                00021790
          ERROR - NO WORKER LEVEL WAS SPECIFIED FOR ACTIVITY CODE ***           00021800
                  TRACE ABOUT DAY ****.****                                     00021810
          JUMP AHEAD                                                            00021820
      *4* PRINT 6 LINES WITH TIME.V THUS                                         00021830
                                                                                00021840
                                                                                00021850
          ERROR - NUMBER OF WORKERS REQUESTED TO PERFORM THE                    00021860
                  ACTIVITY EXCEEDS THE TOTAL NUMBER OF QUALIFIED                00021870
                  WORKERS IN THE SIMULATION.                                    00021880
                  TRACE ABOUT DAY ****.****                                     00021890
        HERE                                                                    00021900
          CALL INFO.PACKAGE(ID)                                                 00021910
          STOP                                                                  00021920
    END                                                                         00021930
```

```
**                                                                    00021940
            ROUTINE INFO.PACKAGE(MZI)                               00021950
CALL MY.LIST(5,MZI)**ACTIVITIES IN ROUTING(MZI)                     00021960
CALL MY.LIST(1)**WORKERS                                            00021970
CALL MY.LIST(6)**MISSIONS IN QUEUE                                  00021980
CALL MY.LIST(7)**MISSIONS IN WAIT FOR MISSION TIME                  00021990
RETURN                                                              00022000
END                                                                 00022010
                              EVENT EO.SIMULATION                   00022020
IF EVENT IS EXTERNAL                                                00022030
  CALL MY.LIST(1)**WORKERS                                          00022040
  CALL MY.LIST(6)**QUEUE                                            00022050
  CALL MY.LIST(7)**WAIT FOR MISSION TIME                            00022060
ALWAYS                                                              00022070
SCHEDULE A FINAL.REPORT NOW                                         00022080
RETURN                                                              00022090
END                                                                 00022100
```

```
**                                                                    00022110
                              ROUTINE TO CLEAR                        00022120
  LET TIME.V = 0 **MUST ZERO BEFORE RESETS FOR VALID STATISTICS       00022130
  RELEASE WNAM, WLEV, WSTA, M.TEAM, WOBSERVED.STATE, MOBSERVED.STATE, 00022140
    TOBSERVED.STATE                                                   00022150
  RELEASE LNIW, LTIT, MIDLE.WORKERS, MIDLE.WORKERS, TIDLE.WORKERS     00022160
**MUST EMPTY SET WHEN ED.SIMULATION WAS AN EXTERNAL EVENT              00022170
  FOR EVERY QMZ IN QUEUE DO                                           00022180
    REMOVE QMZ FROM QUEUE                                             00022190
    DESTROY MISSION CALLED QMZ                                        00022200
  LOOP                                                                00022210
  FOR EVERY XMZ IN WAIT.FOR.MISSION.TIME DO                           00022220
    REMOVE XMZ FROM WAIT.FOR.MISSION.TIME                             00022230
    DESTROY MISSION CALLED XMZ                                        00022240
  LOOP                                                                00022250
  RESET TOTALS OF N.QUEUE                                             00022260
  FOR EVERY LEVEL.OF.WORKER                                           00022270
    RESET TOTALS OF LNIW                                              00022280
  FOR EVERY TEST.PHASE DO                                             00022290
    LET TCNF(TEST.PHASE) = 0                                          00022300
    LET TCMP(TEST.PHASE) = 0                                          00022310
    LET TCNX(TEST.PHASE) = 0                                          00022320
    LET TDCD(TEST.PHASE) = 0                                          00022330
    LET TSTY(TEST.PHASE) = 0                                          00022340
    LET VMCMP(TEST.PHASE) = 0                                         00022350
    LET VMCNF(TEST.PHASE) = 0                                         00022360
    LET VMCNX(TEST.PHASE) = 0                                         00022370
    LET VMCMP(TEST.PHASE) = 0                                         00022380
    LET VMCNF(TEST.PHASE) = 0                                         00022390
    LET VMCNX(TEST.PHASE) = 0                                         00022400
    RESET TOTALS OF TSTY                                              00022410
  LOOP                                                                00022420
  FOR EVERY WORKER                                                    00022430
    RESET TOTALS OF WSTA                                              00022440
  USE UNIT 7 FOR INPUT                                                00022450
```

```
IF DATA IS NOT ENDED                                                    00022460
  FOR I = 1 TO 10                                                       00022470
    LET SEED.V(I) = SAVESEED(I)                                         00022480
ALWAYS                                                                  00022490
FOR J = 1 TO EVT.V DO                                                   00022500
  UNTIL EV.S(J) IS EMPTY DO                                             00022510
    REMOVE THE FIRST K FROM EV.S(J)                                     00022520
    GO TO EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,                    00022530
          CONFIRMED.MISSION, TAKE.BREAK,                                00022540
          ACCOMPLISH.PR1.MISSION, BEGIN.WORKING,                        00022550
          WREPORT, WREPORT OR FINAL.REPORT PER J                        00022560
    *EO.ACTIVITY*                                                       00022570
      DESTROY THE EO.ACTIVITY CALLED K                                  00022580
      GO LOOP                                                           00022590
    *EO.DUTY.HOURS*                                                     00022600
      DESTROY THE EO.DUTY.HOURS CALLED K                                00022610
      GO LOOP                                                           00022620
    *EO.SIMULATION*                                                     00022630
      DESTROY THE EO.SIMULATION CALLED K                                00022640
      GO LOOP                                                           00022650
    *CONFIRMED.MISSION*                                                 00022660
      DESTROY THE CONFIRMED.MISSION CALLED K                            00022670
      GO LOOP                                                           00022680
    *TAKE.BREAK*                                                        00022690
      DESTROY THE TAKE.BREAK CALLED K                                   00022700
      GO LOOP                                                           00022710
    *ACCOMPLISH.PR1.MISSION*                                            00022720
      DESTROY THE ACCOMPLISH.PR1.MISSION CALLED K                       00022730
      GO LOOP                                                           00022740
    *BEGIN.WORKING*                                                     00022750
      DESTROY THE BEGIN.WORKING CALLED K                                00022760
      GO LOOP                                                           00022770
    *WREPORT*                                                           00022780
      DESTROY THE WREPORT CALLED K                                      00022790
      GO LOOP                                                           00022800
```

```
    'MREPORT'                                                    00022810
      DESTROY THE MREPORT CALLED K                               00022820
      GO LOOP                                                    00022830
    'FINAL.REPORT'                                               00022840
      DESTROY THE FINAL.REPORT CALLED K                          00022850
      'LOOP'                                                     00022860
  LOOP                                                           00022870
LOOP                                                             00022880
REWIND 8                                                         00022890
REWIND 9                                                         00022900
PRINT 5 LINES THUS                                               00022910
                                                                 00022920
                                                                 00022930
MODEL HAS BEEN CLEARED FOR NEW INPUT DATA                        00022940
                                                                 00022950
                                                                 00022960
RETURN                                                           00022970
END                                                              00022980
```

```
**                                                                    00022990
                          ROUTINE REPLICATION                         00023000
DEFINE OPTIONAL.SEED AS AN ALPHA VARIABLE                             00023010
FOR EVERY TEST.PHASE DO                                               00023020
  FOR EVERY XFACTOR IN TCANCELLATION.PROBABILITY(TEST.PHASE) DO       00023030
    REMOVE XFACTOR FROM TCANCELLATION.PROBABILITY(TEST.PHASE)         00023040
    DESTROY XFACTOR                                                   00023050
  LOOP                                                                00023060
  FOR EVERY ACTIVITY IN STRUCTURE(TEST.PHASE) DO                      00023070
    REMOVE ACTIVITY FROM STRUCTURE(TEST.PHASE)                        00023080
    DESTROY ACTIVITY                                                  00023090
  LOOP                                                                00023100
LOOP                                                                  00023110
RELEASE TTMC, TNUM, TREQ, TCNF, TCMP, TCNX, TDCD, TDBM, TSTY,         00023120
        N.STRUCTURE, N.TCANCELLATION.PROBABILITY,                     00023130
        M.STRUCTURE, M.TCANCELLATION.PROBABILITY                      00023140
RELEASE KPOA, KTFC, KFWL, KSWL, KTWL, KPMA, KLRL, KURL                00023150
RELEASE 1ARG, 2ARG                                                    00023160
RELEASE SKILL.FACTOR                                                  00023170
RELEASE ONE, TWO, THREE                                               00023180
REWIND 7                                                              00023190
SUBTRACT 1 FROM VRUNS                                                 00023200
USE UNIT 10 FOR INPUT                                                 00023210
READ OPTIONAL.SEED                                                    00023220
IF OPTIONAL.SEED = "YES"                                              00023230
  FOR I = 1 TO 10                                                     00023240
    READ SEED.V(I)                                                    00023250
  PRINT 7 LINES WITH VRUNS-1 THUS                                     00023260
                                                                      00023270
                                                                      00023280
  THE SIMULATION MODEL HAS BEEN RESET FOR A REPLICATION OF THE        00023290
  ENTIRE DATA PACKAGE.  NEW SEED VALUES FOR THE RANDOM NUMBER         00023300
  GENERATORS WERE ENTERED.  THERE ARE ** ADDITIONAL RUNS.             00023310
                                                                      00023320
                                                                      00023330
```

```
  RETURN                                                                    00023340
ELSE                                                                        00023350
PRINT 7 LINES WITH VRUNS-1 THUS                                             00023360
                                                                            00023370
                                                                            00023380
THE SIMULATION MODEL HAS BEEN RESET FOR A REPLICATION OF THE ENTIRE         00023390
DATA PACKAGE.  SEED VALUES FOR THE RANDOM NUMBER GENERATORS WERE            00023400
NOT RESET.  THERE ARE ** ADDITIONAL RUNS.                                   00023410
                                                                            00023420
                                                                            00023430
RETURN                                                                      00023440
END                                                                         00023450
```

APPENDIX D

DATA INPUT INSTRUCTIONS

DATA INPUT INSTRUCTIONS

CODES
A - ALPHA NUMERIC DATA
I - INTEGER DATA
R - REAL DATA
(DECIMAL POINTS MAY BE OMITTED FOR WHOLE NUMBERS)

1. ENTER CONTROL CARD
   $ DATA 07

2. HOW MANY TIMES DO YOU WANT TO RUN THE ENTIRE DATA PACKAGE
   I

3. ENTER THE DATE TO BEGIN THE SIMULATION (MUST BE MONDAY)

MONTH	DAY	YEAR
I	I	I

4. DO YOU WANT WEEKLY REPORTS (NO)
   A

5. DO YOU WANT MONTHLY REPORTS (NO)
   A

6. HOW MANY KIND.OF.ACTIVITY
   I

7. DO YOU WANT TO DELETE THE FEATURE
   REDUCTION.OF.MISSION.EFFECTIVENESS (YES)
   A

8. IF "7" IS YES, ENTER FOR EACH KIND.OF.ACTIVITY.

KOA	ATTRIBUTES						
	KNAM	KPOA	KTFC	KPAR	KFWL	KSWL	KTWL
1	A	I	I	A	I	I	I
2							
-							
M							

NOTE IF AN ACTIVITY CANNOT BE PERFORMED BY THREE DIFFERENT LEVELS OF WORKER, MAKE REPEATED ENTRIES OF A QUALIFIED LEVEL OF WORKER.

9. IF "7" IS NO ENTER FOR EACH KIND.OF.ACTIVITY.

KOA	ATTRIBUTES									
	KNAM	KPOA	KTFC	KPAR	KFWL	KSWL	KTWL	KPMA	KLRL	KURL
1	A	I	I	A	I	I	I	R	R	R
2										
-										
M										

```
10.  HOW MANY TEST.PHASE
     I

     DO 11 THRU 17 FOR EACH TEST.PHASE
     ##################################################
11.  ENTER THE TEST.PHASE NUMBER.
     I

12.  ENTER THE NUMBER OF MISSIONS REQUIRED.
     I

13.  ENTER THE TIME MISSION CONFIRMED (TTMC) RANDOM VARIABLE.
     CUMLA PROB(1) VALUE(1) ---   --- CUMLA PROB(N) VALUE(N)
          R           R                   R          R   *

14.  ENTER THE 1ARG, 2ARG IN MINUTES FOR EACH KIND.OF.ACTIVITY
     1ARG(1)  2ARG(1) ------              -----1ARG(N) 2ARG(N)
        R        R                            R        R
     NOTE   IF AN ACTIVITY IS NOT USED IN A TEST PHASE,
            ENTER ZEROES

15.  ENTER FOR EACH ACTIVITY.
     ACTIVITY          ATTRIBUTES
                     ATPC   ANOA   ANWR
        1             I      I      I
        2
        -
        M                          *
     NOTE   SINCE ATPC INDICATES THE ORDER IN WHICH ACTIVITIES
            ARE PERFORMED, HIGHER NUMBERS SHOULD BE ASSIGNED AS
            ATPC'S WITHIN A TEST PHASE AND IN SUCEEDING TEST
            PHASES.  SEE TEXT FOR ASSIGNMENT OF PRIORITIES.

16.  ENTER * TO DELETE MISSION CANCELLATION FEATURE.
     *

17.  IF "16" IS NOT * ENTER THE PROB OF CNX FOR EACH FACTOR.
     R                                              R *
     ##################################################

18.  ENTER OK
     A

19.  ARE THERE ANY EMPERICAL FUNCTIONS  (NO)
     A

     #####     #####     #####     #####     #####     #####
```

```
20.  IF "19" IS YES,
       HOW MANY INTEGER.STEP.FUNCTION
       I
       ENTER FOR EACH FUNCTION
       FUNCTION                                  PAIRS
          CUMULA  PROB(1) VALUE(1)---CUMULA  PROB(N) VALUE(N)
        1       R                  I       R                I
        2
        -
        M

21.  IF "19" IS YES,
       HOW MANY REEL.STEP.FUNCTION
       I
       ENTER FOR EACH FUNCTION
       FUNCTION                                PAIRS
          CUMULA  PROB(1) VALUE(1)---CUMULA  PROB(N) VALUE(N)
        1       R                  I       R                I
        2
        -
        M

22.  IF "19" IS YES,
       HOW MANY CONTINUOUS.FUNCTION
       I
       ENTER FOR EACH FUNCTION
       FUNCTION                                PAIRS
          CUMULA  PROB(1) VAULE(1)---CUMULA  PROB(N) VALUE(N)
        1       R                  R       R                R
        2
        -
        M
     #####     #####     #####     #####     #####     #####

23.  HOW MANY DUTY DAYS PER WEEK
     I

24.  WILL YOU ALLOW ACTIVE MISSIONS ON THE WEEKENDS  (YES OR NO)
     A

25.  ENTER THE TIMES TO START AND STOP WORKING.
     VBDDT  VBB1T  VEB1T  VBLBT  VELBT  VBB2T  VEB2T  VEDDT
       R      R      R      R      R      R      R      R

26.  ENTER THE NUMBER OF DAYS PREPARATION REQ FOR AN ACTIVE MSN.
     R
```

```
27.  ENTER OK
     A

     REPEAT 28 THRU 36 FOR EACH SIMULATION RUN.
     ##########################################################
28.  ENTER * TO DELETE TRACE MESSAGE FEATURE.
     *

29.  IF 28 IS NOT * ENTER DAY TO START AND STOP TRACE MESSAGES.
     R   R *

30.  ENTER OK
     A

31.  ARE ALL SKILL FACTORS THE SAME   (YES)
     A

32.  HOW MANY LEVEL.OF.WORKER
     I

33.  HOW MANY WORKERS
     I

34.  ENTER FOR EACH WORKER                          # IF "31" WAS NO,
     WORKER                      ATTRIBUTES          SKILL FACTORS
                                WNAM      WLEV   #  KOA(1)-----KOA(N)
       1                         A          I        R          R
       2                                           #
       -
       N                                           #
     NOTE   WORKERS SHOULD BE ENTERED IN ORDER BY LEVEL

35.  ENTER THE NUMBER OF DAYS BETWEEN MSN FOR EACH TEST.PHASE
        R   ----                    -----   R
     NOTE   TO SCHEDULE MORE THAN 1 MISSION PER DAY, ENTER THE
            VALUE ((END DUTY DAY - BEGIN DUTY DAY)/24)/NUMBER
            EXAMPLE   OFFICE HOURS 0800-1700, 3 MISSIONS/DAY
                      ((17.0-8.0)/24)/3 = 0.125

36.  ENTER OK
     A
     ##########################################################
37.  ENTER CONTROL CARD
     $      DATA    08
```

```
38.  ENTER EXTERNAL CONFIRMED.SCHEDULED.MISSION
     EVENT   DATE   TIME   TEST.PHASE   SEQUENTIAL ACTIVITY CODES
     A       I      I      I            I -- SEE NOTE -- I*
     NOTE    IF AN ACTIVITY CODE IS NOT PART OF THE INTERNAL
             EVENT SEQUENCE DESCRIBED IN 15, ENTER AFTER THAT
             ACTIVITY CODE.
      ANOA    APOA    ANHR    AFHL    ASHL    ATHL
       I       I       I       I       I       I

39.  ENTER CONTROL CARD
     $       DATA    09

40.  ENTER EXTERNAL EO.SIMULATION
     EVENT      DATE      TIME
      A          I         I *

     41 THRU 43 ARE REQUIRED FOR REPLICATIONS.
     ##########################################################
41.  ENTER CONTROL CARD.
     $       DATA    10


42.  IF SPECIFIED SEED VALUES FOR THE RANDOM NUMBER GENERATORS
     ARE TO BE USED, ENTER YES AND THE VALUES. (10 VALUES EACH)
     1  A I I I I I I I I I I
     2
     -
     M

43.  IF THE CURRENT SEED VALUES ARE TO BE USED, ENTER NO FOR
     EACH REPLICATION.
     A ----     ---- A
     ##########################################################

44.  ENTER CONTROL CARD
     $      ENDJOB
```

APPENDIX E

SAMPLE DATA

DATA INPUT

```
$:DATA:07
1
5 24 1976
NO
NO
14
NO
PMSNPG 2 9 YES 1 2 3   0  0   0
PMSNPA 2 9 YES 1 2 3   0  0   0
CGDMSN 1 9 NO  1 2 3  10 10  20
CAMSN  1 9 NO  1 2 3  30 30 100
PSTMNG 3 9 NO  1 2 3   0  0   0
PSTMNA 2 9 NO  1 2 3   0  0   0
DATARD 1 9 NO  3 3 3   5 10  20
DATAAN 3 9 YES 3 1 2   0  0   0
PMSNI  2 9 YES 1 1 1   0  0   0
CMSNI  1 9 NO  1 1 1  10 10  15
PSTMNI 3 9 NO  1 2 2   0  0   0
MSNNEL 1 3 NO  1 1 1   0  0   0
ANTAPE 3 9 YES 3 1 2  10 10  20
ANLOGS 3 9 YES 3 1 2   0  0   0
8
1
6
0 8 .75 10 1.0 12 *
360 480 0 0 120 180 0 0
  0   0 0 0   0   0 0 0 0 0
  0   0 0 0   0   0 0 0 0 0
211 1 3
212 3 3 *
*
2
12
0 10 .75 13 1.0 14 *
  0   0   0   0   0   0 0 0
  0   0   0   0   0   0 0 0
480 600 120 180 120 180 0 0 0 0 0 0
221   9 2
222 10 2
223 11 2 *
.1 .2 .1 *
3
4
0 10 .75 13 1.0 14 *
0 0 90 120 0 0 105 135
  0  0  30  90   0   0 0 0
  0  0   0 0 0   0   0 0 0 0 0 0
```

```
231 2 3
232 4 2
233 6 3 *
.1 .2 .1 *
4
22
0 10 .75 18 1.0 21 *
  0   0  30 120     0     0  240  300
  0   0 120 180 11520 14400 1440 2160
  0   0   0   0     0     0    0    0
360 480 120 150
241  2 3
242  4 6
243  6 4
244 13 2
246 14 1
247  7 0
248  8 1 *
.2 .2 *
5
7
0 10 .75 18 1.0 21 *
  0   0  30 120     0    0 240 300
  0   0 120 180 1440 2160 480 600
  0   0   0   0    0    0   0   0
360 480 120 150
151  2 3
152  4 3
153  6 3
154 13 2
155 14 1
156  7 0
157  8 2 *
.2 .1 *
6
1
0 7.5 1.0 7.5 *
0 0 0 0 0 0    0 0
0 0 0 0 0 0    0 0
0 0 0 0 0 0 8640 0
0 0 0 0
261 12 2 *
.1 .1 *
7
4
0 10 .75 13 1.0 14 *
0 0  30 120     0    0 180  240
0 0 120 180 1440 4320 720 2880
0 0   0   0    0    0   0    0 0 0 0 0
271 2 4
272 4 4
```

```
273 6 4
274 / 0
275 8 3 *
.1 .1 *
8
10
0 10 .75 13 1.0 14 *
0 0  30 120    0    0 180 240
0 0 120 180 2880 5768 360 480
0 0   0   0    0    0   0   0 0 0 0 0
281 2 3
282 4 3
283 6 3
284 7 0
285 8 1 *
.1 .1 *
OK
NO
5
NO
7.5 9.75 10.0 11.5 12.5 14.75 15.0 16.0
1.0
OK
*
OK
NO
3
9
TE1 1   1   1 1 1   1 1 1.5 2   1 1 1 1 1.5 1.5
TE2 1   1   1 1 1   1 1 1.5 2   1 1 1 1 1.5 1.5
PO  1   1   1 1 1   1 1 1.5 2   1 1 1 1 1.5 1.5
TT1 2   1   1 1 1 1.5 1 1.5 2   1 1 1 1   2   2
TT2 2   1   1 1 1 1.5 1 1.5 2   1 1 1 1   2   2
TT  2   1   1 1 1 1.5 1 1.5 2   1 1 1 1   2   2
TA1 3 1.5 1.5 1 1   1 1   1 1 1.5 1 1 1   1   1
TA2 3 1.5 1.5 1 1   1 1   1 1 1.5 1 1 1   1   1
OA  3 1.5 1.5 1 1   1 1   1 1 1.5 1 1 1   1   1
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
OK
$:DATA:08
CONFIRMED.MISSION 8/16/76 9 30 4 241 242 243 244
245 13 3 3 3 1 2 247 248 *
CONFIRMED.MISSION 8/30/76 10 00 6 262 8 3 3 3 3 3 *
CONFIRMED.MISSION 1/1/84 9 30 *
$:DATA:09
EO.SIMULATION 1/1/84 10 30 *
$:DATA:10
NO NO NO NO NO NO NO NO NO
$:ENDJOB
```

APPENDIX F

SAMPLE OUTPUT

ATTRIBUTES OF EACH KIND OF ACTIVITY

ACT	KNAM	KPOA	KTFC	KPAR	KFWL	KSWL	KTWL	KPMA	KLRL	KURL
1	PMSNPG	2	9	YES	1	2	3	0.	0.	0.
2	PMSNPA	2	9	YES	1	2	3	0.	0.	0.
3	CGDMSN	1	9	NO	1	2	3	10.00	10.00	20.00
4	CAMSN	1	9	NO	1	2	3	30.00	30.00	100.0
5	PSTMNG	3	9	NO	1	2	3	0.	0.	0.
6	PSTMNA	2	9	NO	1	2	3	0.	0.	0.
7	DATARD	1	9	NO	3	3	3	5.00	10.00	20.00
8	DATAAN	3	9	YES	3	1	2	0.	0.	0.
9	PMSNI	2	9	YES	1	1	1	0.	0.	0.
10	CMSNI	1	9	NO	1	1	1	10.00	10.00	15.00
11	PSTMNI	3	9	NO	1	2	2	0.	0.	0.
12	MSNNEL	1	3	NO	1	1	1	0.	0.	0.
13	ANTAPE	3	9	YES	3	1	2	10.00	10.00	20.00
14	ANLOGS	3	9	YES	3	1	2	0.	0.	0.

##### NUMBER	MISSION TIME PROB	TIME	ACTIVITY SEQUENCE AND RQMTS CODE	ACTIVITY	WORKERS REQ
8	0.	10.00	281	2	3
	.75	13.00	282	4	3
	1.00	14.00	283	6	3
			284	7	0
			285	8	1

ATTRIBUTES OF TEST PHASE

TNUM	TREQ	N.STRUC	N.CX.PROB
8	10	5	2

ATTRIBUTES OF EACH ACTIVITY IN STRUCTURE(TEST PHASE)

ATPC	ANOA	APOA	ATFC	AICT	ANWR	APAR	AFWL	ASWL	ATWL
281	2	2	9	0.	3	YES	1	2	3
282	4	1	9	0.	3	NO	1	2	3
283	6	2	9	0.	3	NO	1	2	3
284	7	1	9	0.	0	NO	3	3	3
285	8	3	9	0.	1	YES	3	1	2

ATPC	ACEL	APMA	ALRL	AURL	M.STRUCTURE	N.TEAM
281	1.00	0.	0.	0.	1	0
282	1.00	30.00	30.00	100.0	1	0
283	1.00	0.	0.	0.	1	0
284	1.00	5.00	10.00	20.00	1	0
285	1.00	0.	0.	0.	1	0

ATTRIBUTES OF EACH XFACTOR IN TCANCELLATION.PROB(TEST PHASE)

XUPL	M.TCANCELLATION.PROBABILITY
.10	1
.10	1

ATTRIBUTES OF EACH WORKER

WORKER	WNAM	WLEV	WSTA	M.TEAM
1	TE1	1	3	0
2	TE2	1	3	0
3	PO	1	3	0
4	TT1	2	3	0
5	TT2	2	3	0
6	TT	2	3	0
7	TA1	3	3	0
8	TA2	3	3	0
9	OA	3	3	0

THE SKILL FACTOR OF EACH WORKER

	WORKER								
	1	2	3	4	5	6	7	8	9
ACT									
1	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
2	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.50	1.50	1.50	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7	1.50	1.50	1.50	1.50	1.50	1.50	1.00	1.00	1.00
8	2.00	2.00	2.00	2.00	2.00	2.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	1.50	1.50	1.50	2.00	2.00	2.00	1.00	1.00	1.00
14	1.50	1.50	1.50	2.00	2.00	2.00	1.00	1.00	1.00

FINAL TEST FORCE REPORT
(FOR SIMULATION ENDING 10 18 1976)

MISSION STATISTICS
FOR TEST PHASE 1 ARE

NUMBER SCHEDULED----------	=	6	MISSIONS
NUMBER CANCELLED----------	=	0	MISSIONS
NUMBER REQUIRED----------	=	6	MISSIONS
EFFECTIVE PRODUCTIVITY------	=	6.0000	MISSIONS
AVERAGE MISSION DURATION----	=	1.4120	DAYS
STD DEV--------------------	=	.6779	DAYS
TOTAL OF DURATION TIMES-----	=	8.4718	DAYS
SUM OF SQ OF DURATION TIMES-	=	14.72	DAYS
DATE COMPLETED------------	=	6 1 1976	

MISSION STATISTICS
FOR TEST PHASE 8 ARE

NUMBER SCHEDULED----------	=	13	MISSIONS
NUMBER CANCELLED----------	=	2	MISSIONS
NUMBER REQUIRED----------	=	10	MISSIONS
EFFECTIVE PRODUCTIVITY------	=	10.3721	MISSIONS
AVERAGE MISSION DURATION----	=	7.7844	DAYS
STD DEV--------------------	=	2.1568	DAYS
TOTAL OF DURATION TIMES-----	=	85.6289	DAYS
SUM OF SQ OF DURATION TIMES-	=	717.75	DAYS
DATE COMPLETED------------	=	10 18 1976	

NOTE SINCE THE DURATION OF CANCELLED MISSIONS IS ZERO, STATISTICS ARE COMPUTED FOR COMPLETED MISSIONS ONLY.

QUEUE REPORT

MAX LENGTH	AVG LENGTH	STD DEV	SUM OF Q TIMES
12.0 MSN	3.1854 MSN	3.3276 MSN	470.4 DAYS

HISTOGRAM OF QUEUE LENGTH

QUEJE LENGTH	STATE PERCENTAGE
0 MISSIONS	32.22 %
1 MISSIONS	15.92 %
2 MISSIONS	8.94 %
3 MISSIONS	3.98 %
4 MISSIONS	4.45 %
5 MISSIONS	4.77 %
6 MISSIONS	4.71 %
7 MISSIONS	8.89 %
8 MISSIONS	8.02 %
9 MISSIONS	4.81 %
10 MISSIONS	2.94 %
11 MISSIONS	.34 %
12 MISSIONS	.01 %

NOTE ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS. PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING MAY NOT SUM TO 100.

Section 5. Queue Report

HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL

STATE TIME
(DAYS)

NUMBER OF IDLE WORKERS	LEVEL 1	LEVEL 2	LEVEL 3
0	135.9906	127.8727	132.5936
1	3.4398	2.6188	2.7880
2	.6038	1.9215	1.1470
3	7.6358	15.2571	11.1415
4	0.	0.	0.
TOTAL IDLE TIME	27.55	52.23	38.51

ANALYSIS OF WORKER STATUS

HOURS IN STATE
STATE PERCENTAGES

WORKER	IDLE	BUSY	BREAK	OFF DUTY	OVERTIME
TE1	224.30 HRS	536.42 HRS	140.28 HRS	2399.41 HRS	243.67 HRS
	6.33 %	15.14 %	3.96 %	67.70 %	6.88 %
TE2	218.95 HRS	546.45 HRS	135.60 HRS	2289.53 HRS	353.55 HRS
	6.18 %	15.42 %	3.83 %	64.60 %	9.98 %
PO	218.06 HRS	548.40 HRS	134.53 HRS	2272.20 HRS	370.88 HRS
	6.15 %	15.47 %	3.80 %	64.11 %	10.46 %
TT1	418.86 HRS	336.08 HRS	146.06 HRS	2555.10 HRS	87.98 HRS
	11.82 %	9.48 %	4.12 %	72.09 %	2.48 %
TT2	418.28 HRS	334.17 HRS	148.56 HRS	2557.67 HRS	85.41 HRS
	11.80 %	9.43 %	4.19 %	72.17 %	2.41 %
TT	416.45 HRS	335.99 HRS	148.56 HRS	2556.09 HRS	86.99 HRS
	11.75 %	9.48 %	4.19 %	72.12 %	2.45 %
TA1	307.34 HRS	441.65 HRS	152.01 HRS	2616.78 HRS	26.30 HRS
	8.67 %	12.46 %	4.29 %	73.84 %	.74 %
TA2	308.95 HRS	440.04 HRS	152.01 HRS	2613.27 HRS	29.81 HRS
	8.72 %	12.42 %	4.29 %	73.74 %	.84 %
OA	307.86 HRS	440.88 HRS	152.26 HRS	2619.81 HRS	23.27 HRS
	8.69 %	12.44 %	4.30 %	73.92 %	.66 %

VITA

James Paul Eggers was born on 12 March 1946 in Centralia, Illinois. He graduated from Centralia High School in 1964 and attended the University of Illinois from which he received the degree of Bachelor of Science in Electrical Engineering in February 1969. Upon graduation he received a commission in the USAF through the ROTC program. He was employed as an engineer for the Illinois Bell Telephone Company until called to active duty in February 1969.

After completing pilot training at Laredo AFB, Texas, he served 18 months in Southeast Asia as a member of the 361st Tactical Electronic Warfare Squadron. He served as an instructor pilot and flight examiner in the 4410th Special Operations Group at England AFB, Louisiana, and as a training monitor and air operation officer in the 1st Special Operations Wing at Hurlburt Field, Florida. He entered the Air Force Institute of Technology in May 1975 to earn the degree of Master of Science in Systems Analysis.

Mailing address: 219 S. Cherry Street
Centralia, Illinois 62801

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GOR/SM/76D-7	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) JOINT TEST FORCE SIMULATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James P. Eggers Captain		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN)		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS 3246 Test Wing/TG Eglin AFB, Florida 32542		12. REPORT DATE December, 1976 13. NUMBER OF PAGES 180
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JERRAL F. GUESS, Captain USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Manpower
Requirements
Simulation
Test Methods

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This study presents the background, development, and use of a generalized event-oriented simulation model of work activities. The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive? The program is written in SIMSCRIPT II.5 and is adaptable to a particular joint test force by using

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

free-format data inputs.

Any test force which is responsible for the performance of activities may be studied using this model. Any number of activities, missions, test phases, workers, and levels of workers may be defined through the input data. Detailed input instructions are provided to tailor the model to the analyst's requirements. Numerous options are available to include: worker skill factors, mission reduction or cancellation probabilities, activity performance capability with reduced personnel, reports, parametric studies, and replications of the entire data package.

The function of each routine and event is discussed and a complete program listing is appended. A sample study is provided to illustrate many of the significant features.